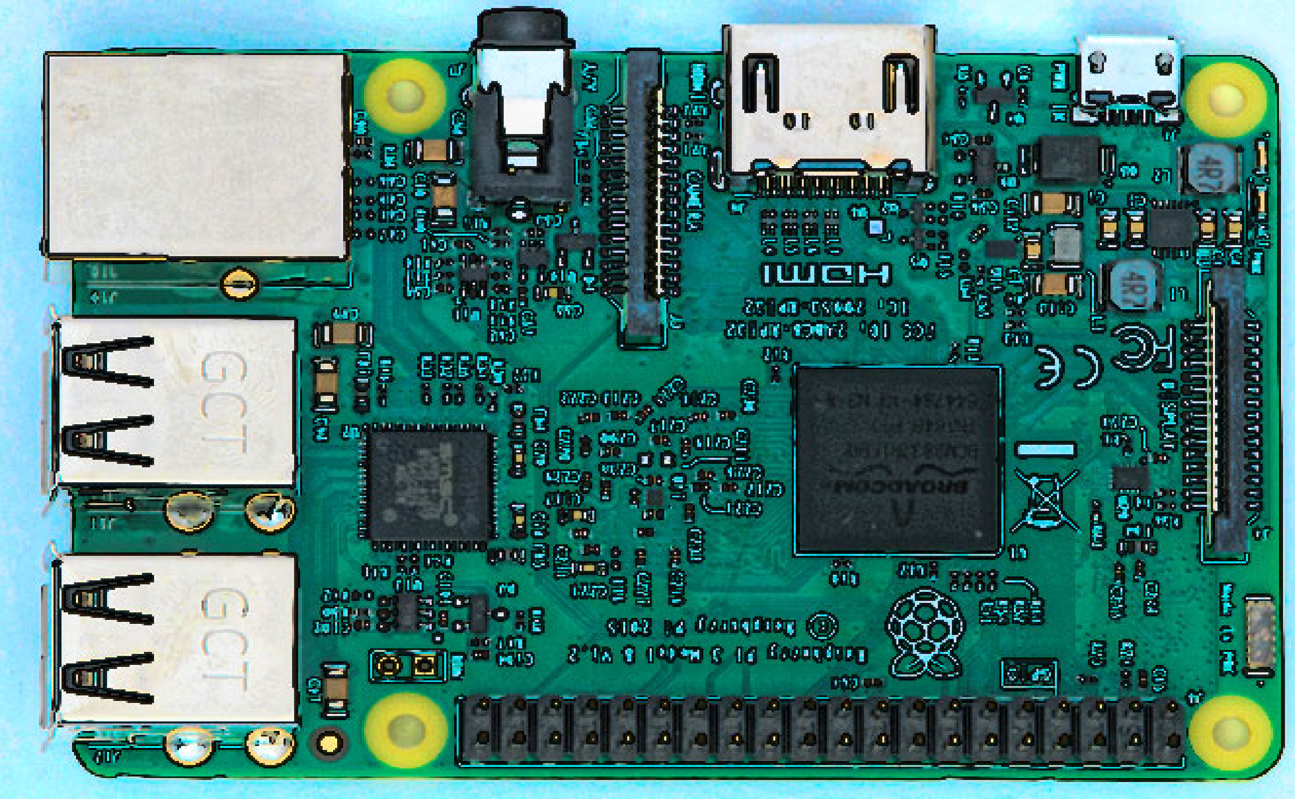
Einar Krogh

An Introduction to the Internet of Things



bookboon LEARNING

EINAR KROGH

AN INTRODUCTION TO THE INTERNET OF THINGS

An Introduction to the Internet of Things
2nd edition
© 2022 Einar Krogh & bookboon.com
ISBN 978-87-403-3956-7

CONTENTS

PREFACE

This book is written for introductory courses on embedded systems and the Internet of Things (IoT). It consists of two parts. The first part introduces embedded systems. The central topics are sensors, actuators, and the architecture of embedded systems. The second part of the book is about the IoT. It introduces the IoT, the architecture of the IoT and important IoT technologies. Other topics are also included as cloud computing and Big Data in connection with the IoT.

In the time to come, the IoT will have a greater and greater impact on our lives. The IoT is also very popular, and many predict that it will be the next big digital revolution.

I would like to thank Professor Øystein Haugen and Doctoral Research Fellow Marius Geitle for reading the manuscript for this book and for their many suggestions.

Proof-Reading-Service.com
PhD theses, journal papers, books and other professional documents

Proof-Reading-Service.com Ltd, Devonshire
Business Centre, Works Road, Letchworth Garden
City, Hertfordshire, SG6 1GJ, United Kingdom
Office phone: +44(0)20 31 500 431
E-mail: enquiries@proof-reading-service.com
Internet: http://www.proof-reading-service.com
VAT registration number: 911 4788 21
Company registration number: 8391405

PART 1 EMBEDDED SYSTEMS

1 INTRODUCING EMBEDDED SYSTEMS

1.1 WHAT IS AN EMBEDDED SYSTEM?

An embedded system is a combination of hardware and software to perform a specific task. Allowing software to control hardware provides the opportunity for intelligent behaviour and smart solutions. An example of the use of an embedded system is one that regulates the lighting in a room. The light turns on by itself when someone is in the room, and it turns off when no one is in the room. Another example of an embedded system is one that automatically irrigates plants when the soil is dry, without the need for human involvement.

An embedded system consists of three main components:

- Hardware,
- Application software and
- Operating system.

An operating system is computer software that manages hardware and other software. While hardware and software are always included in an embedded system, there are exceptions when it comes to the operating system. Very simple embedded systems do not always have an operating system.

Software for embedded systems is usually referred to as firmware. Instead of storing data on a hard drive as on computers, individual programs in an embedded system are normally stored on a chip and called firmware.

Figure 1.1 The development of embedded systems has become much easier since the introduction of two platforms: Arduino and Raspberry Pi. The figure shows a Raspberry Pi 4 Model B.

Electronic equipment designed for the engineering market is classified as an embedded system. This electronic system can be programmed to operate and to organise one or more tasks. Embedded systems are an important part of today's electronic industry.

1.2 REAL-TIME EMBEDDED SYSTEMS

Real-time systems are computer systems that monitor, respond to, or control an external environment. The external environment is connected to an embedded system that often has sensors, actuators, and other interfaces.

A real-time system must be able to respond to events in an external environment as soon as they happen. Another name for real-time systems is reactive systems because their primary purpose is to respond to events in the environment.

An example of a real-time system is an embedded system that brakes a car when there is something on the road in front of it. The car must brake immediately (in real time) if an accident is to be avoided. Figure 1.2 lists other examples of real-time embedded systems.

There are two important features of a real-time embedded system:

1. Real-time embedded systems must perform flawlessly accurate calculations for events.
2. The response to events must occur very quickly during a predefined time interval.

In real-time systems where real-time calculations are required with accurate results to be delivered within a short time span, the operating system usually plays an important role. With the growing complexity of the hardware in embedded systems, there is a need for an operating system that meets system requirements and does not miss deadlines.

Systems for cars, subways, aircraft, railways and ships
Traffic control for motorways, airspace, railway tracks and ship routes
Videoconference applications
Medical systems for radiation therapy and patient monitoring
Military applications such as firearms, tracking, command and control
Robot production systems
Telephone, radio and satellite communication
Computer games

Figure 1.2 Some examples of real-time embedded systems

Real-time embedded systems are everywhere. Today's systems range from kitchen appliances and car management systems to control systems for air traffic, military weapons facilities, and production line control, including robotics and automation.

1.3 EMBEDDED SYSTEMS COMPARED WITH COMPUTERS

In contrast to a computer designed to do many tasks, embedded systems are most often designed to perform only one task – for example, regulating a traffic light. Many embedded systems are also real-time systems, which ordinary computers are not.

Because embedded systems only have one specific purpose, they are more limited in hardware and software functionality than a computer. In the case of hardware, this may mean less processing power, less power consumption, less memory, less hardware functionality and so on. In terms of software, limitation can mean fewer applications, simpler applications and no or a small operating system. Because embedded systems only focus on a particular task, they can be made more cheaply and more efficiently than a computer.

Desktop Computer	Embedded System
Runs different programs at different times depending upon the needs of the user.	Always runs a single, dedicated application.
Has large amounts of random-access memory and disk space; both can be readily and cheaply expanded if required.	Has sufficient memory, but not an excess. Adding more memory is difficult.
All PCs have an essential identical hardware architecture and run identical software. Software is written for speed.	Embedded systems are highly variable with different central processing units, peripherals, operating systems, and design properties.
Boot up time may be measured in minutes and the operating system is loaded from a disk and initialised.	Boot up is almost instantaneous, measured in seconds.

Figure 1.3 Some differences between embedded systems and a computer

The differences between computers and embedded systems can describe the nature of an embedded system. Simply put, an embedded system is any data system found in hardware equipment that is not a computer.

1.4 FEATURES OF EMBEDDED SYSTEMS

An embedded system is a system that has a specific purpose and performs either a single or only a few operations. Most embedded systems do not have a user interface for humans, but some may have a kind of user interface such as a touch screen. There are also embedded systems that have quite complex user interfaces, such as mobile phones.

A typical feature of embedded systems is low power consumption and low cost. This often results in limited processing possibilities.

An embedded system demands high quality and reliability. Embedded systems are often used in hardware that should function flawlessly for many years. For example, if technical equipment used in an operation in a hospital breaks down, this could have life-threatening consequences, or if a system that controls a car fails on the highway, there could be an accident. Figure 1.4 lists the common features of embedded systems.

Requirements	
Processing Power	Microcontrollers or microprocessors control embedded systems.
Operating System	The embedded operating system must be reliable and capable of running with restrictions on memory, size, time, and processing power.
Memory	Computer programs designed for embedded systems are handled as firmware and are stored in read-only memory or on flash memory chips.
PowerConsumption	Power consumption is an important factor for any embedded battery-powered system. It is the size of power consumption that determines the life of the battery.
Flexibility	Flexibility is the ability to change system functionality without investing extra costs. Software is considered flexible if it can be updated at any time with a new version.
Size	An embedded system should preferably be as small as possible.
Reliability	An embedded system should be highly reliable to achieve good performance over its lifetime.
Security	If there is system failure, there should be no damage to the components.
Maintenance	It is important that an embedded system can be repaired or replaced as soon as possible – that is, within a specified time interval.

Figure 1.4 Common features of embedded systems

1.5 ROBOTS AND EMBEDDED SYSTEMS

A robot is a mechanical and programmed device designed to execute one or more tasks automatically with speed and precision. In the old days, robots were controlled by large and expensive computer systems. Many of these robots were stationary because they were too heavy to carry around. Mobile robots had to be connected to a computer via cables or wires, but connection and response speed was problematic. The development of embedded systems has had great significance for robots because it has solved these problems. Figure 1.5 shows an example of a robot.

A robot consists of three main components:

1. Sensors that provide feedback from an external environment,
2. A mechanical unit (actuator) that can perform actions on the environment and
3. An embedded system for communicating between a mechanical device and the sensory data.

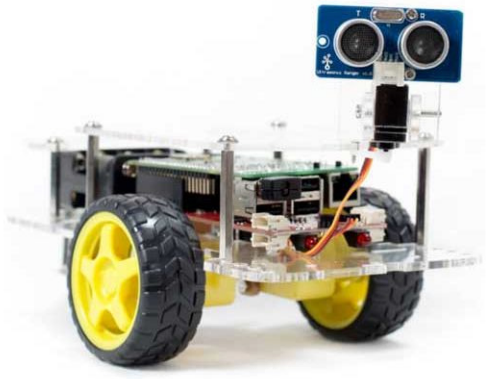


Figure 1.5 An example of an embedded system. The GoPiGo is a mobile robot car controlled by a Raspberry Pi card, which contains program code controlling the robot's movements (image from Dexter Industries)

Very few robots look like the metallic robots in science fiction. Any mechanism that has actuators, sensors and a controller can be classified as a robot. The definition also includes units that we would not consider as robots, such as the block-free braking system of newer cars. Figure 1.6 gives some examples of robots.

Types of Robots	Use
Industrial	Material handling, welding, inspection, improving productivity, laboratory applications
Collaborative	Interacts safely and effectively with human workers while performing simple industrial tasks
Mobile	Move around on legs, tracks, or wheels
Educational	Features learning platforms that are geared to teaching robotics to children, students, and amateurs
Domestic	Vacuum cleaners, floor washing robots, ironing robots, etc.
Military	Use of robots for military combat; smart missiles and autonomous bombs can be considered robots

Figure 1.6 Some types of robots

2 USE OF EMBEDDED SYSTEMS

Embedded systems are everywhere: washing machines, microwave ovens, household appliances, digital watches, toys, traffic lights, industrial robots, agricultural machinery, and many others. Embedded systems are used in consumer electronics that include mobile phones, video game consoles, digital cameras, global positioning satellite (GPS) receivers and printers. Home automation uses embedded systems to control light, sound, climate, security, and monitoring. Embedded systems are used in transportation, fire protection, medical applications, and life-critical systems, as these systems can be isolated from hacking and thus be more reliable. For fire protection, the embedded systems can be designed to have greater ability to withstand higher temperatures and thus continue to function if a fire.

Aircrafts and car transport systems are increasingly using embedded systems. New aircrafts include advanced technology that also has high security requirements.

2.1 EMBEDDED SYSTEMS IN VEHICLES

Use of advanced embedded systems in automobiles has increased rapidly in the past decades. More and more equipment is being changed from mechanical systems to electronic systems. An embedded system is important in a vehicle's electronic system because of its versatility and flexibility. Every year, automobile manufacturers pack more embedded systems into their cars for different functionalities like ignition, security, and audio. The aim is to make the vehicle more efficient and safer.

We shall look at some applications of embedded systems in cars.

Adaptive Cruise Control

If someone in the late 1900s had told us that there would soon come a new technology that will end car accidents, no one would have believed it. However, embedded systems that support the driver make driving much safer.

Many modern cars have an embedded adaptive cruise control system (ACC), a control system for road vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead. It determines the car's speed using a braking system that considers the distance between the vehicle it is in and the vehicle in front. Such cars are usually equipped with radar or light detecting and ranging (LIDAR) to determine distance.

Lane Centring

Lane centring, also known as auto steer, is a mechanism designed to keep a car centred in the lane, relieving the driver of the task of steering. Together with ACC, this feature can allow driving without a driver.

Airbag Control System

All modern cars have airbags to make driving safer. To inflate the airbags at the right time, you have an airbag control system. It detects a collision using a crash sensor and sends a signal to the airbags so that they are inflated. The entire process from start to finish takes 0.1 s.

Anti-Lock Braking System

An anti-lock braking system (ABS) is designed to control vehicle braking in a way that gives less of a chance to slip on slippery roads. An ABS operates by preventing the wheels from locking up during braking, thereby maintaining tractive contact with the road surface. It ensures better contact with the road by controlling brake pressure if a car begins to slip while braking.

Other Uses of Embedded Systems in a Car

Other uses of in-vehicle systems include an embedded navigation system (ESC), electronic stability program (ESP), traction control (TCS), tyre air pressure control, automatic four-wheel drive, electronic fuel injection, embedded rain-sensing system and embedded-based automatic car parking system (Figure 2.1).

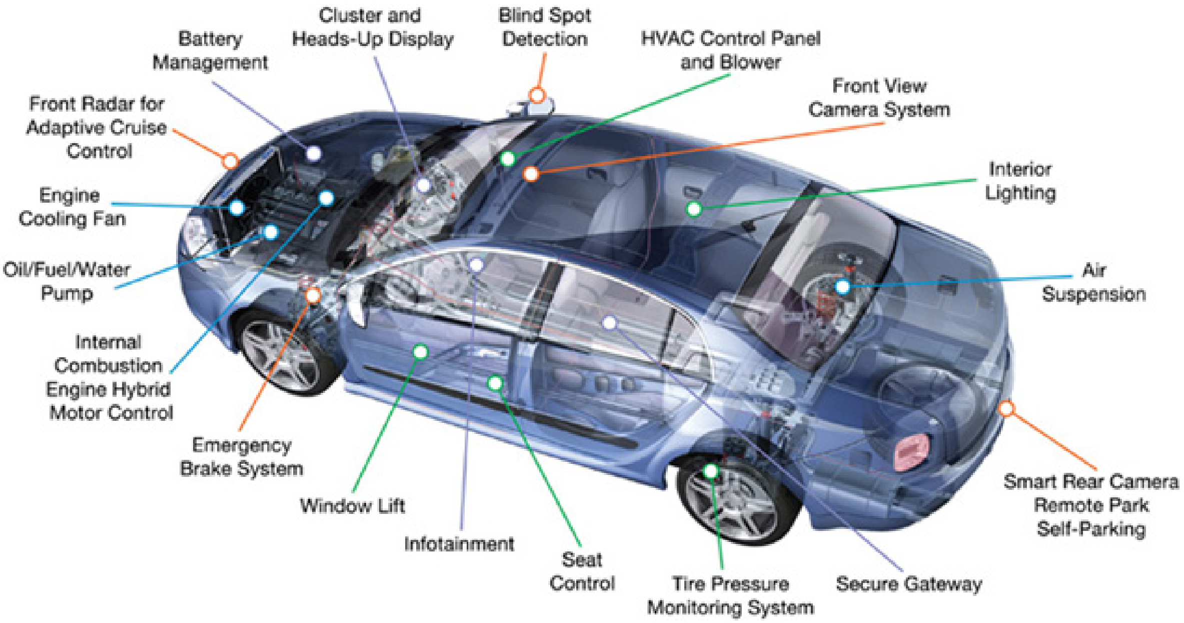
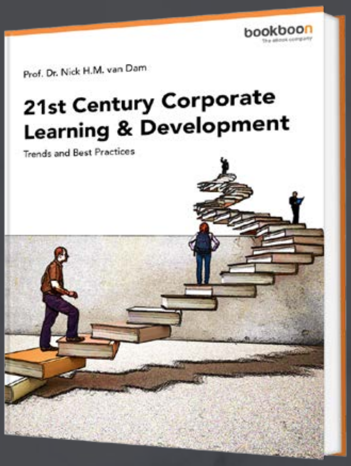

Battery Management
Cluster and Heads-Up Display
Blind Spot Detection
HVAC Control Panel and Blower
Front View Camera System
Front Radar for Adaptive Cruise Control
Interior Lighting
Engine Cooling Fan
Oil/Fuel/Water Pump
Air Suspension
Internal Combustion Engine Hybrid Motor Control
Emergency Brake System
Smart Rear Camera Remote Park Self-Parking
Window Lift
Infotainment
Seat Control
Tire Pressure Monitoring System
Secure Gateway

Figure 2.1 Today's automobiles contain dozens of embedded systems

Free eBook on
Learning & Development

By the Chief Learning Officer of McKinsey

Download Now


bookboon
Prof. Dr. Nick H.M. van Dam
21st Century Corporate Learning & Development
Trends and Best Practices

2.2 SOME EXAMPLES OF EMBEDDED SYSTEMS

Embedded systems have become a part of our daily lives. Figure 2.2 includes some examples.

Washing machines and dishwashers
Lighting systems
Refrigerators and freezers
Vacuum cleaners
Mobile phones
Smart watches and digital watches
Air conditioners and thermostats
Electric cookers and coffee machines
Electronic parking meters and parking services
CD players, iPods and MP3 players
Home security systems
Fire alarms and carbon monoxide detectors
Printers, copiers, fax machines and scanners
Digital cameras
Electronic safes
GPS navigation devices
Heart rate monitors and pacemakers
Wi-Fi routers
Electronic toys

Figure 2.2 Examples of items with embedded systems

3 SENSORS AND ACTUATORS

A sensor detects and responds to input from the physical environment. Sensor's sense, which is to say they act as the eyes and ears of an embedded device and detect changes in the environment around them.

An actuator accepts a control command and produces a change in the physical system by generating force, motion, heat, flow and so on. You may say actuators are the hands of embedded systems.

Both sensors and actuators have many practical applications.

3.1 TRANSDUCERS

A transducer is any physical device that converts one form of energy to another. Transducers that convert physical quantities into mechanical quantities are called mechanical transducers. Transducers that convert physical quantities into electrical quantities are called electrical transducers.

Some examples of transducers are:

- An electric motor converts electricity into mechanical energy or motion,
- A speaker converts electrical signals to sound waves,
- An incandescent lamp produces light by converting electrical energy into optical energy and
- A solar cell converts light into electricity.

Transducers allow technical equipment to interact with the physical environment. A transducer must therefore have a processing unit and a communication interface.

A sensor is a kind of a transducer. A sensor transforms a physical phenomenon into an electrical impulse that can be used of a technological system. For example, a microphone is a transducer (sensor) that converts sound waves into electrical signals.

3.2 SENSORS

Sensors are key components of the IoT. They are important in the work of monitoring processes, measurements, and data collection. A sensor is a device that detects events or changes in the environment and sends information about it to another location – for example, to a server or a web page. Most sensors are designed to measure a physical quantity and then transform it into a digital value that can be read by humans or used by some data system.

Sensors are the reason why the IoT is constantly growing. There are many different types of sensors on the market, and they are used for various purposes that reach all aspects of human life.

Sensor technology is being developed faster and faster because of new discoveries in materials and nanotechnology. The result is increased accuracy, reduced size and cost, and the ability to measure or to find values that have not previously been possible. In fact, sensor technology is developing so rapidly and becoming so advanced that we will see billions of new sensors in production annually within a few years.

The low price of most sensors helps keep the IoT expense down and allows the use of embedded systems on a large scale. The IoT has changed the manufacturing industry, and sensors are central to the use of IoT.

Use of Sensors

Sensors are used widely and there are hundreds of sensors that measure different things: temperature, flow, voltage, humidity and so on. For example, autonomous vehicles are full of sensor technology and have sensors to measure power, load, torque, motion, speed, displacement, position, vibration, and shock.

Sensors used in production equipment in a factory can help identify bottlenecks in a manufacturing process. By tackling bottlenecks, waste in production time is reduced.

Instead of standard preventive maintenance, which means machine maintenance, predictable maintenance refers to using sensors to predict quite accurate when machines need maintenance.

Sensors and actuators must be reliable. In complex embedded systems, a fault in a sensor or an actuator can trigger catastrophic events. Detection of sensor and actuator errors can be difficult, and it affects the performance of critical systems.

Storage of Data from Sensors

Deciding whether to store data from sensors locally or in the cloud is often a dilemma. There are advantages and disadvantages to both options.

Speed is one of the main advantages of local storage. Storing data on external hard drives is faster than uploading it to the cloud. You also have full control of your backups, which means better control of who accesses your data. Disconnecting the drives from the network makes your data safe from attacks.

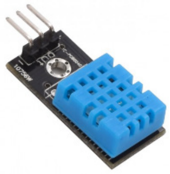
There are many advantages for backing up sensor data to the cloud. For one, it is cost-effective, and maintenance is not an issue because cloud storage providers handle all the upgrades and troubleshoot any issues that arise. Another advantage of cloud storage is scalability. When you need to increase storage space, it is as simple as notifying your service provider. You can increase or decrease space as needed. Should a disaster occur on-site, your data will remain safe. Securing data remotely means there is no worry of losing backups of your data. Accessibility is also a plus of cloud storage. Data stored in the cloud is easily accessed on any device that has an Internet connection. You can log into your cloud account and your data are there when you need it.

A disadvantage with cloud storage is security and privacy. There are concerns with valuable and important data being stored remotely. Before adopting cloud technology, you should be aware that you are giving sensitive information to a third-party cloud service provider, and this could potentially be a risk. Another disadvantage is lifetime costs. With public cloud storage, the costs over the years might increase and tend to add up. This is when the lifetime costs will hit you. If your applications are local and your data are in the cloud, then it can add to networking costs.

3.3 SOME SENSORS

Sensors perform very different tasks, and each IoT system requires a specific type of sensor. We will look at some sensors and their use.

Temperature Sensors



This widely used sensor type measures the temperature or the heat of a given medium. There are several types of temperature sensors on the market. Temperature sensors are used in everything from simple thermostats to highly sensitive semiconductors that can control complex processes.

Proximity Sensors

A proximity sensor can determine the distance to a nearby object without having physical contact with the object. Proximity sensors use electromagnetic radiation or radar to detect movements or obstructions.

Proximity sensors are good to detect movement. They are a common component of equipment that involves safety, security, or efficiency. These sensors are therefore used in vehicles to detect obstacles in front of the vehicle when in motion.

Proximity sensors are used in stores. Dealers use proximity sensors to investigate to which goods customers get closest – and are thus most interested in – in their stores. The data are processed and sent to the dealers' mobile phones.

Proximity sensors can be used in parking systems, museums, and airports, among other places. For example, parking sensors are proximity sensors designed for vehicles to alert the driver of obstacles with parking. A ground proximity warning system (GPWS) is a system to alert pilots if their aircraft is in immediate danger of flying into the ground or an obstacle.

Motion Detector Sensors

A motion detector is an electronic device used to detect physical motion in an area and to transform motion into an electrical signal. The motion can be the movement of an object or the movement of people.

Motion detection plays an important role in the security industry. Companies use these sensors in areas where there should always be no movement. It is then easy to notice if anyone is moving with these sensors installed. They are used, for example, for intrusion detection, door control, automatic parking systems, automated sinks, toilet fans, hand dryers and automated lighting. Although their primary use is now in the security industry, the number of possible applications of these sensors will only grow as the use of motion detection technology develops.

Light Detecting and Ranging

LIDAR is a type of sensor that measures the distance to a target by measuring a laser pulse reflection on the target. LIDAR broadcasts laser energy. As a laser hits an object, some of the energy will be reflected towards the LIDAR transmitter. This type of active sensing machine is also capable of analysing anything that crosses its path. LIDAR is now used in automated and self-driving vehicles, robotics, surveillance, and agriculture.

Water Quality Sensor

Water quality sensors are used to estimate water quality and to monitor ions primarily in water distribution systems. Water is used practically everywhere. These sensors therefore play an important role as they monitor the quality of water used for various purposes. They are in use in several industries.

Optical Sensors

Fibre optic sensor technology is used to detect electromagnetic energy such as light, electricity and similar elemental particles. They can send, receive, and convert light energy into electrical signals.

Fibre optic sensors are used in energy, health care, aviation, chemical and environmental IoT platforms. Optical sensors can be ideal for environments such as oil refining, mining operations, pharmaceutical production, and chemical treatment. We can expect marked growth of fibre optic sensors as a part of the increase in industrial applications in automation, as they are considered very suitable sensors for the IoT.

Chemical Sensors

Chemical sensors are used in several different industries. The goal is to indicate changes in fluid or in the air. They play an important role in larger cities, where it is necessary to record chemical changes to protect the population. An important use of chemical sensors is in industrial environmental monitoring and process control, intentionally or accidentally released harmful chemical detection, explosive and radioactive detection, space station recycling processes, the pharmaceutical industry, and laboratories, among others.

Level Sensors

A sensor used to determine the level or amount of liquids or other substances flowing in an open or closed system is called a level sensor. Level sensors measure the level of fluids. They can be used for smart waste management and recycling purposes. Other applications include measurement of tank levels, diesel meters, high or low-level alarms and irrigation control. Common uses of level sensors are fuel gauges and fluid levels in open or closed containers, sea and tsunami monitoring, water reservoirs, medical equipment, compressors, hydraulic reservoirs, machine tools, beverages, pharmaceutical treatment and high- or low-level detection, among many others.

Infrared Sensors

An infrared (IR) sensor measures IR light radiating from an object nearby. IR light has several applications. It can help doctors monitor blood flow in humans, visualise heat leakage in houses and identify environmental chemicals in the environment.

IR sensors are now used in a variety of IoT projects, especially in the health care system as they facilitate blood flow and blood pressure monitoring. They are even used in several common smart devices, such as smartwatches and smartphones. Other common applications include household appliances and remote controls, breathing analysis, IR visualisation (i.e. visualising heat leakage in electronics, monitoring blood flow, for art historians to see under layers of paint), usable electronics, optical communication, non-contact temperature measurements and blind angle detection. IR sensors will play an important role in the smart home industry, as they have a wide range of applications.

Image Sensors

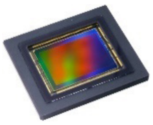


Image sensors are devices used to convert images into electronic signals for display or storage to file. Digital cameras, medical imaging, night vision equipment, thermal imaging equipment, radar and sonar make extensive use of image sensors.

One of the most well-known uses includes the automotive industry, where images play a crucial role. With these sensors, the system can recognise signs, obstacles, and many other things that a driver would generally notice on the road. They play a very important role in the IoT industry, as they directly affect the progress of self-driven cars. They are also used in security systems, where images help to capture details of a perpetrator.

3.4 SENSOR FUSION

We can get information from many different sensors. If we get information about an event from more than one sensor, it may be advantageous to combine the different pieces of information. Sensor fusion is the ability to bring together inputs from multiple sensors to form a single model or image of the environment. Sensor fusion is the combination of sensory data derived from disparate sensors such that the resulting information has less uncertainty than would be possible when these sources were used individually.

Sensor fusion could reveal more about the context than a single sensor can provide. This is important in the IoT space because a single thermal sensor has no notion on what causes a rapid thermal change. With time-correlated data from multiple sensors, processing can make better decisions based on more data.

There are different methods for sensor fusion. Some such methods are the central limit theorem (CLT), Kalman filter, Bayesian networks and convolutional neural networks.

3.5 ACTUATORS

Actuators represent a different type of transducer. An actuator works in the opposite direction of a sensor. It takes electrical signals and converts them into physical action. An example of an actuator is an electric motor that creates movement.

An actuator is a mechanism that is responsible for moving or controlling something. An actuator requires a control signal and an energy source. The energy source may be electric current, hydraulic fluid pressure or pneumatic pressure. When the control signal is received, the actuator responds by converting the energy into mechanical motion.

Electric Actuator

An electric actuator is a device that can create movement. A motor that converts electrical energy into mechanical energy powers an electric actuator. It is a clean and easily accessible form of actuator because it does not use oil or other fossil fuels directly.

Hydraulic Actuator

A hydraulic actuator consists of a cylinder or fluid motor that uses hydraulic power to facilitate mechanical operation. The mechanical movement produces an effect that can be a linear, rotating, or oscillating motion. As liquids are difficult to compress, a hydraulic actuator can exert great force.

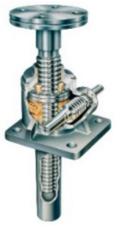
Pneumatic Actuator

Pneumatics is the utilisation of energy by means of compression and expansion of gases. A pneumatic actuator converts energy generated by a vacuum or compressed air to either a linear or a rotary motion. Pneumatic energy can react quickly at the start and stop because there is no need for a power source in the reserve for operation. Pneumatic actuators make it possible to produce considerable forces from relatively small pressure changes.

Thermal or Magnetic Actuator

Actuators that can be activated by thermal or magnetic energy have been used in commercial applications. Thermal actuators are often compact, lightweight, economical and have high power.

Mechanical Actuator

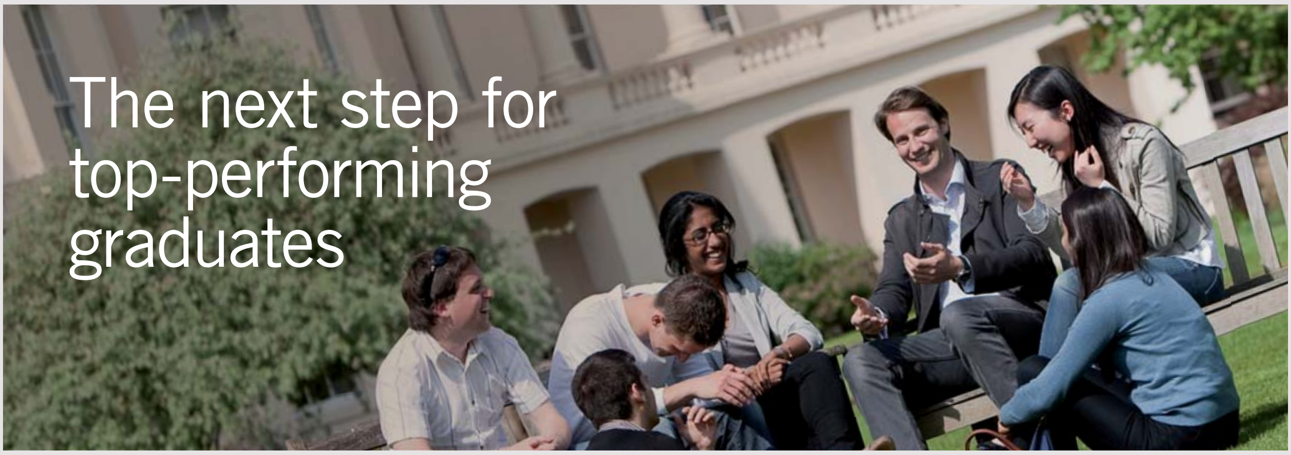


A mechanical actuator performs motion by converting motion, such as rotational motion, to another type of motion, such as linear motion. An example is a gear that runs around driving a vehicle moving straight ahead. Operation of mechanical actuators is based on combinations of structural components, such as gears and rails, or pulleys and chains.

3.6 USE OF ACTUATORS

Actuators are devices that convert some type of stored energy into motion. Embedded systems with electric motors can create movement. Electric motors convert electrical energy into mechanical energy so that they can perform environmental operations. By using an actuator, you get the opportunity to perform many different tasks – for example, robot control, home aids such as watering flowers, camera control, unmanned aircraft, three-dimensional (3D) writing control and more.

An electric motor often is used to create rotation around a fixed axis that can be used to drive wheels, pumps, belts, and robot arms. There are three types of engines commonly used for rotation: servomotors, direct current (DC) motors and stepper motors.



Masters in Management

London
Business
School

Designed for high-achieving graduates across all disciplines, London Business School's Masters in Management provides specific and tangible foundations for a successful career in business.

This 12-month, full-time programme is a business qualification with impact. In 2010, our MiM employment rate was 95% within 3 months of graduation*; the majority of graduates choosing to work in consulting or financial services.

As well as a renowned qualification from a world-class business school, you also gain access to the School's network of more than 34,000 global alumni – a community that offers support and opportunities throughout your career.

For more information visit www.london.edu/mm, email mim@london.edu or give us a call on +44 (0)20 7000 7573.

* Figures taken from London Business School's Masters in Management 2010 employment report

Servomotor

A servomotor is an actuator that allows precise control of position, speed, or acceleration. It consists of a motor connected to a sensor that provides feedback on the position. Servomotors are controlled by sending an electrical pulse that determines how large the movement is. Servomotors are small and highly energy efficient. These features allow them to be used to control remote- or radio-controlled toy cars, robots, and aircraft.

Servomotors are used in applications such as robotics, computer numerical control (CNC) machinery or automated manufacturing. CNC is the automated control of machining tools (drills, boring tools, lathes) by means of a computer. Servomotors are also used in industrial applications, robotics, in-line manufacturing, pharmaceuticals, and food services.

Direct Current Motors

A DC motor converts electrical energy to mechanical energy. The DC motor is the most common actuator used in electronic projects. DC motors are used in many contexts from toys to advanced robots. They are ideal motors to use when there is a need for continuous rotation as well as the wheels of an electric vehicle.

DC motors are cheap and easy to use. They also come in a large selection of sizes to accommodate different tasks. DC motors convert electrical energy into mechanical energy. The speed of rotation can be adjusted by the size of the power supply. Low power supply provides low rotation, and high-power supply provides rapid rotation.

Stepper Motors

Stepper motors are DC motors that move with fixed steps. Stepper motors share a full rotation in a series of equal steps. The motor will rotate one step at a time.

Stepper motors rotate a certain angle such as 1.8 degrees. This means that each time it receives a power pulse, it will rotate 1.8 degrees. This ability allows stepper engines to rotate quite accurately, with a rotation error of less than 5%. With a computerised control, you can therefore achieve very precise positioning and/or speed regulation. Hence, stepper motors rotate in a different way than DC motors, which rotate continuously, based on the amount of power supplied.

Linear Actuator

A linear actuator creates movement in a straight line, as opposed to the circular motion of a conventional electric motor. Linear actuators are used in machine tools, industrial machines and in peripherals, such as disk drives and printers, in valves and dampers and in many other places where linear motion is required.

4 ARCHITECTURE OF EMBEDDED SYSTEMS

Every embedded system has an architecture because an embedded system is composed of different components – software and hardware – that work together. An architecture comprises these components and the relationship between them.

If you want to design an embedded system, you must be familiar with the architecture of the equipment you want to create. It is important to plan the design for embedded systems to avoid mistakes or an expensive outcome. Understanding the architecture of an embedded system is necessary for good system design. All elements of an embedded system must interact with each other. Without understanding this interaction, it will be difficult to understand how the embedded system will behave under different conditions in the real world. An embedded system is made up of three components (Figure 4.1).

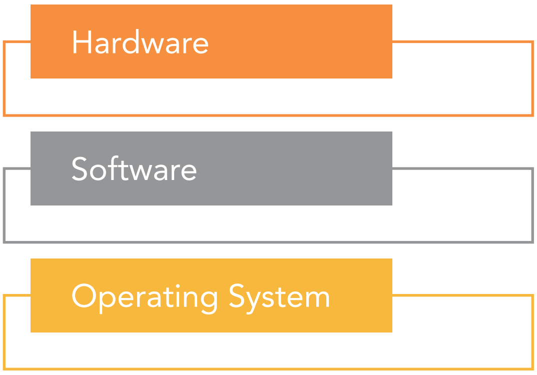


Figure 4.1 Main components of an embedded system

We will look at the architecture of each of these components.

4.1 HARDWARE ARCHITECTURE

Embedded systems consist of electronic equipment placed on a circuit board. The main components of such a board are a processor, memory, data buses and input/output devices. Two different architectures for how these components work together are von Neumann architecture and Harvard architecture.

Von Neumann Architecture

Von Neumann architecture is a way of designing computers where both program instructions and data reside in the same memory device. The processor is separate from the memory device. The mathematician John von Neumann developed this type of architecture in 1945. He suggested an architecture consisting of the following components:

1. A memory that should contain both data and instructions in binary form.
2. A processing unit that could perform mathematical and logical operations.
3. A controller that interprets instructions in memory and ensures that they are executed.
4. Input and output devices that provide communication between user and control unit.

Von Neumann architecture supports simple hardware. It allows the use of a single memory. It also has a small memory (Cache) near the processor. Von Neumann architecture is used in personal computers, laptops, and workstations. All modern computers use this architecture.

Harvard Architecture

The Harvard architecture use different memory devices for program instructions and data. Instructions and data also use different data buses. This allows the processor to access both instructions and data at the same time.

In a system with a pure von Neumann architecture, instructions and data are stored in the same memory, so instructions and data are retrieved over the same data bus. This means that a central processing unit (CPU) cannot read an instruction and perform data storage at the same time.

In a computer that uses the Harvard architecture, the CPU can both read an instruction and perform a data storage access simultaneously. Using Harvard architecture will thus be faster than using von Neumann architecture. The Harvard architecture is used in digital signal processors and microcontrollers. A typical feature of microcontrollers is that they have limited software and memory for data, and they benefit from the Harvard architecture for fast processing while simultaneously accessing instructions and data.

Microprocessors and Microcontrollers

Embedded systems are based on microprocessors or microcontrollers. Both types are designed to perform calculations.

Microprocessors have a slightly simpler construction than microcontrollers because the microprocessor consists only of one CPU and thus requires the connection of other components as well as memory chips. Microprocessors are used in various areas of technology. For example, they are present in mobile phones. They are also used in MP3 players, refrigerators, microwaves, some remote controls, printing devices and GPS receivers, among others.

Microcontrollers, on the other hand, are designed as independent devices. Microcontrollers not only have a CPU but also memory and external devices such as flash memory, random access memory (RAM) or serial communication port. Most microcontrollers in use today are embedded in other machinery, such as automobiles, robots, telephones, medical equipment, household appliances and peripherals for computer systems.

Modern embedded systems are often based on microcontrollers and numerous microcontrollers have been developed for embedded systems use. General-purpose microprocessors are also used in embedded systems, but they generally require more support circuitry than microcontrollers.

Advanced RISC Machine Processor

ARM is short for advanced RISC machine. The ARM architecture is used in many products because it is small, relatively inexpensive to produce and has low power consumption. Due to its low power consumption, an ARM processor is well suited for use in portable devices. Almost all modern mobile phones and personal digital assistants contain ARM processors. This makes them the most widely used 32-bit microprocessor family in the world. ARM technology currently accounts for over 75% of all 32-bit embedded processors and is used in 98% of mobile phones sold every year.

In tablets, ARM processors provide longer battery life because they use less power. ARM processors also produce less heat than larger Intel processors, allowing tablets to be thinner. While not as fast as Intel PC or portable processors, ARM processors still provide reasonable speed, especially for mobile computing.

4.2 SOFTWARE ARCHITECTURE

An embedded system is usually designed to perform one specific task. Performing only one simple task does not require large resources. The software in an embedded system is therefore often designed for the following:

1. Little available memory,
2. Low processor speed and
3. Minimal power consumption.

Different embedded systems can be designed for very different tasks. Therefore, there are several types of software architecture for embedded systems. We shall look at different types of software architecture.

Simple Control Loop

In this design, the program consists of a single loop. The loop calls a function that performs some task. This design is therefore called either a simple control loop or just a control loop.

Interrupt-Controlled System

Some embedded systems are controlled by interrupts. This means that a specific event will call software that performs a task on the system. This type of system is used if events need a short time to be executed and if event handling is simple. These systems usually also run a simple task in a main loop, but this task can wait a little bit if an unexpected event occur.

Cooperative Multitasking

A multitasking system resembles the simple control loop system, except that the program system is designed to perform multiple tasks, and each task has its own run environment. The advantages and disadvantages of cooperative multitasking are the same as for the control loop, except that it is easier to add new software.

Pre-emptive Multitasking or Multi-threading

Pre-emptive multitasking is a type of multitasking that allows computer programs to share operating systems and underlying hardware resources. It divides the overall operating and computing time between processes, and the switching of resources between different processes occurs through predefined criteria.

Simple Operating System Kernel

The kernel is a computer program at the core of a computer's operating system that has complete control over everything in the system. The usual functioning is that the operating system kernel allocates memory and switches different running threads in and out of the CPU. Processes in user mode implement key functions such as file systems, network interfaces, etc.

Embedded Systems with a Large Operating System Core

In this case, a relatively large operating system core is adapted to an embedded system. This gives programmers an environment like a computer operating system such as Linux or Microsoft Windows and is therefore very suitable for development. The downside is that it requires significantly more hardware resources. It is often more expensive, and the complexity of these cores can lead to less predictability and reliability.

Examples of an embedded operating system core are Embedded Linux and Windows IoT. Despite the increased cost of hardware, this type of embedded system has increased in popularity, especially on the more powerful embedded devices such as wireless routers and GPS navigation systems.

Additional Software Components

In addition to the core operating system, many embedded systems have additional components in the upper layer. These components consist of network protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP), Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), File Transfer Protocol (FTP) and Controller Area Network (CAN). They can also contain storage functions such as File Allocation Table (FAT) and have flash memory systems.

If the embedded device has audio and video features, the current drivers will be present in the system. As far as the monolithic cores are concerned, many of these components are included.

4.3 OPERATING SYSTEM ARCHITECTURE

An operating system is software that manages computer hardware and software resources and provides common services for computer programs. If you want to make an embedded system, it must have an operating system. Very simple embedded systems can do without an operating system, but it is rare for embedded systems not to have one.

Embedded systems often use operating systems designed specifically for embedded use. For example, all mobile phones use an operating system made for mobile phones. The operating system handles the user interface and all the basic functions of the phone.

An embedded operating system is designed to be efficient and reliable. Efficiency often comes at the expense of losing some functionality. An embedded operating system has fewer features than a standard computer operating system. The embedded operating system is often adapted to the embedded system. Many of the usual operating system components are removed, as they are not needed.

The hardware that runs an embedded operating system is often limited in terms of resources such as memory. The operating systems have a limited task adapted to run a particular program that performs a particular operation. To take advantage of the processing power of the CPU, software developers frequently implement critical code they write into the operating system. This machine effective language can potentially result in better speed and performance at the expense of portability and maintenance. Embedded operating systems are most often written in a system programming language such as C.

Embedded operating systems can either be operating systems designed specifically for the embedded device or one of the many operating systems adapted to run on top of an embedded system. Common embedded operating systems include Symbian, Windows Phone, Windows IoT and Linux.

Embedded Operating Systems Versus Computer Operating Systems

An important difference between most embedded operating systems and computer operating systems is that the former is a part of the operating system, often so that the entire software is only a single executable file. Indeed, the system can usually only run a single program. Unlike PC operating systems, an embedded operating system is unable to load and to execute various applications.

Because embedded operating systems often run only one application, hardware has little memory, and a slow CPU is typically used. Embedded operating systems are typically programmed in machine language to really benefit from the limited computing resources. This means that the operating system is adapted to the hardware for which it was designed, and this operating system will not be compatible with other hardware systems with other configurations.

Commercial Operating Systems

There are many operating systems to purchase on the market. These operating systems have both advantages and disadvantages compared with free operating systems. There are many commercial real-time operating systems, and many are from well-established and reputable suppliers. However, buying one of these is something that should be carefully considered. The company's size, product quality and use are important factors.

An important requirement is the possibility of technical support. When buying an operating system, both the buyer and the seller make a long-term commitment. One side of the relationship is the consideration of possible CPU migration in the future. A well-established provider of real-time operating systems can deliver new versions of the operating system, and their product is probably designed to simplify upgrades. Good documentation is important and can be expected from a commercial real-time operating system vendor.

One disadvantage of commercial operating systems is that, technically, each embedded system is different. CPU, memory, and external devices vary from device to device. Moreover, the operating system must fit the embedded system. Commercial operating systems also require licenses.

Free Operating Systems

Free operating systems are often easily downloadable, real-time operating systems that are quite popular. Linux is not quite a free operating system because a supported version of Linux is not free. However, a supported and packaged version of Linux is something on which most embedded developers are likely to spend money.

The advantage of free operating systems is that you do not have to pay anything for the operating system, nor will you have to do so later, as there are no license fees. Free operating systems would often include the source code, which is useful for reference as the documentation may be limited and it may be difficult to get support later. It is also a requirement for configuration and transfer to a new hardware environment.

A disadvantage of free operating systems is that implementing an operating system on an embedded device is a long-term commitment, so the issue of long-term support is important. For a free operating system, you cannot rely on long-term support.

4.4 MIDDLEWARE

Middleware is software that provides common services and capabilities to applications outside of what is offered by the operating system. Middleware helps developers build applications more efficiently. It acts like the connective tissue between applications, data, and users. Middleware can be defined as any system software that is not a part of the operating system kernel, the device drivers, or the user applications. Although middleware is not a part of the operating system, some operating systems can integrate middleware into the operating system.

In an embedded system, middleware is system software that is usually located either on device drivers or on the top of the operating system and can sometimes be part of the operating system itself (Figure 4.2).

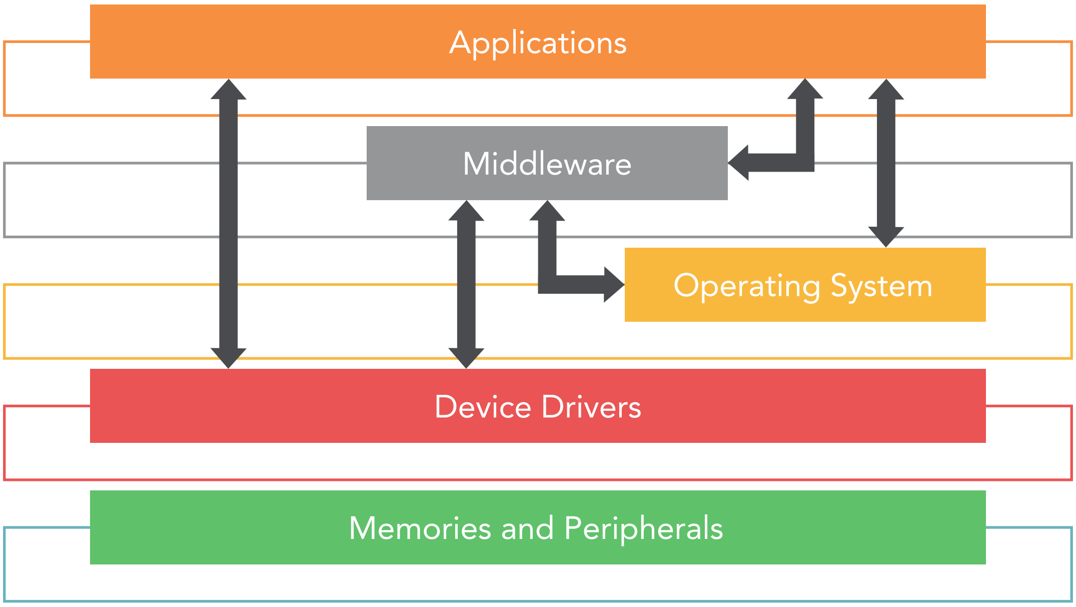


Figure 4.2 The embedded system architecture

Middleware is usually software that lies between applications and the core or driver software. Middleware can also be software that serves other software. More specifically, middleware is an abstraction layer commonly used on embedded systems with two or more applications to provide flexibility, security, portability, connectivity, interconnection, and application collaborative mechanisms.

An important advantage of using middleware is that it can reduce the complexity of the applications by centralising the software infrastructure. However, using middleware in a system can affect scalability and performance. In short, middleware affects the embedded system in all layers.

4.5 SOME OPERATING SYSTEMS FOR EMBEDDED SYSTEMS

The following are some operating systems designed for embedded systems.

Embedded Linux

Linux can be used as an operating system in embedded systems. The benefits of using Linux as the basis for an embedded operating system include the following: supplier independence, low cost, open source, and hardware support. The developer is Community Linus Torvalds.

Windows IoT

Windows 10 IoT Core is built for small, secure smart devices and supports ARM processors. With all the power of Windows, Windows 10 IoT shares all the benefits of developing Windows systems worldwide. The developer is Microsoft.

Tiny OS

TinyOS is an embedded operating system and a platform for wireless devices that use low power. It is an open-source operating system, licensed for low power wireless devices. It is used in sensor networks, personal networks, smart buildings, and smart meters. The developer is TinyOS Alliance.

Contiki

Contiki is an operating system for network-based systems focusing on low energy used in wireless devices in the IoT. This open-source operating system is highly portable and supports multitasking for embedded systems in memory-efficient networks and in wireless sensor networks. Contiki is designed to run on types of hardware devices that are severely constrained in memory, power, processing power and communication bandwidth. The developer is Adam Dunkels.

Nano-RK

Nano-RK is a fully pre-emptive real-time operating system (RTOS) with network support for use in wireless sensor networks. Nano-RK supports fixed-priority multitasking to ensure that task times are met, along with support for the CPU, network, sensors, and actuators. The developer is Carnegie Mellon University.

LiteOS

LiteOS is an open and interactive Unix-like operating system designed for wireless sensor networks. With the tools provided with LiteOS, you can operate one or more wireless sensor networks in a Unix-like manner, transfer data, install applications, retrieve results, or configure sensors. You can also develop applications for nodes and distribute such programs wirelessly to sensor nodes. LiteOS is open source, and the developer is Huawei Technologies Co., Ltd.

QNX

QNX is a commercial Unix-like real-time operating system, aimed primarily at the embedded systems market. QNX was one of the first commercially successful microkernel operating systems. It is used in a variety of devices including cars and mobile phones. The developer is Blackberry.

4.6 STANDARDS OF EMBEDDED SYSTEMS

Some of the key components of embedded systems are represented by specific procedures called standards, which dictate how components should be made and which other components the system needs to function satisfactorily. Standards are determined by specific organisations. The Institute of Electrical and Electronics Engineers Standards Association (IEEE SA) is an operating unit within the IEEE that develops global standards in a broad range of industries, including artificial intelligence (AI) systems, IoT, consumer technology and electronics, learning technology, information technology (IT) and robotics, telecommunication, transportation, home automation and many more. IEEE SA has developed standards for over a century.

Standards can determine the functionality of all three layers in the model for the architecture of an embedded system. They can be classified as market-specific standards, general-purpose standards or standards that can belong to both classifications. Most market-specific standards, except networks and some TV standards, are often made for special groups of embedded systems. General-purpose standards, however, are often made for a market of embedded systems.

Programming language standards are an example of a general-purpose standard. A programming language can be used in various embedded systems. Network functionality standards can be implemented in all equipment that uses network communication. Standards classified as market specific define a functionality that belongs to a particular group of embedded systems. Some examples of market specific standards are described below.

Consumer Electronics

This group includes equipment used by consumers. There may be personal digital assistants (PDAs); TVs; games; toys; home appliances such as microwave ovens, dishwashers and washing machines; and Internet-enabled equipment.

Medical Equipment

This category comprises instruments, apparatuses, devices, articles used alone or in conjunction with any other software included. These pieces of equipment can be used for diagnosis, prevention, reading, treatment, or control of physical condition of patients.

Industrial Automation and Control

This group includes robotic equipment such as sensors, control of movement, human/ machine communication equipment and industrial switches.

Networking and Communication

This equipment connects endpoints in networks, routers, hubs, and switches. This market also includes equipment used for audio–video transmissions.

5 PROGRAMMING OF EMBEDDED SYSTEMS

The following describes some programming languages that are popular in programming of embedded systems.

Many embedded systems are written in C or C++.

C is a good choice for embedded system development. C combines the low-level functionality of the assembly language very neatly with modern-day programming conventions. Porting embedded programs across different devices is much easier than programs written in most other languages. It can be used on almost any advanced embedded system platform that exists.

C++ is an object-oriented language based on C. If you want to develop slightly larger program systems, C++ may be preferable to C. The ability to use overloaded functions and constructors makes C++ an ideal choice for embedded systems programming.

Embedded C++

Embedded C++ is a subset of the C++ programming language aimed to make embedded systems. The language includes only the parts of C++ that are used heavily in the embedded system community and omits key C++ features like exception handling, multiple inheritance, namespaces, templates, and virtual base classes. Any standard C++ compiler can be used to compile embedded programs written in Embedded C++. Embedded C++ tries to avoid excessive memory consumption by removing most C++ core functionalities that are not exclusively used in embedded systems programming.

Python

Python has gradually become popular with embedded systems. Python is in many ways a flexible language. What makes Python good for programming is its readability. The design specifications for the language emphasise the importance of readable code and compact, elegant syntax. Python may not be as useful for embedded programming as C or C ++, but with many available libraries, it is easy to implement functions. It is excellent for automation of testing and data collection and analysis.

Java is widely used to develop embedded systems. Java is an object-oriented language that is highly portable. Java makes it much easier to write extensible, portable, and downloadable embedded system applications. A wide array of developer tools and powerful libraries make Java a suitable choice for embedded systems programming.

Go was developed by Google and is available for a variety of processors and platforms. Go is an open-source programming language that makes it easy to build simple, reliable, and efficient software.

Go comes with built-in features for unit testing, thus making testing your embedded application very easy. The rich associated programming interface (API) documentation of this embedded systems programming language is beneficial for both new and veteran developers alike.

Go adds an explicit hash table type, as well as types that can be very useful for collecting data from and sending data to separate sensors and actuators. The ability to process a network of sensors and devices is supported.

JavaScript

Programmers who make software for embedded systems are often familiar with scripting. They can often choose a scripting language because it is a fast way to solve problems.

JavaScript sounds like a variant of Java, but the two are very different. The two languages are similar in that there are, for example, some libraries that can be used by both, but the languages have been developed separately and share no syntax or semantics.

The massive array of developer tools and third-party libraries makes JavaScript a suitable choice for developing fast and reliable embedded software. The event-driven, functional programming paradigm employed by JavaScript can be utilised to build stable embedded systems easily.

B# is a small, object-oriented programming language designed to run multiple threads. B# is designed specifically for small-embedded systems. B# is like C#, but many of the features of C# that are not required for embedded projects are removed in B#.

B# was designed from the ground up as a small, highly effective embedded control language. B# supports real-time control features. The embedded virtual machine allows B# to run on a variety of platforms. It uses only 24K memory, much less than what is needed for many of the other languages used.

C# is widely popular for building enterprise software. However, this innovative programming language is also used heavily for developing embedded systems for industrial purposes. With its strongly typed, component-oriented programming style, C# encompasses many useful features for embedded system programming. Moreover, as Microsoft maintains this embedded system programming language, you can easily find tons of documentation on it.

Rust

Rust is one of the most modern programming languages being used to develop embedded systems. Rust combines the benefits of low-level languages such as C and C++. From small microcontrollers to powerful single board computers, Rust allows you to port your embedded system's code easily across a wide range of systems. Rust offers great community support.

Forth

Forth is a language designed and optimised for embedded system programming. It is a stack-oriented language and is primarily used for system-level programming. A language that has existed since the 1970s, Forth is still used today in many embedded systems (small-computerised devices) because of its portability, efficient memory use, short development time and fast execution speed.

Verilog

Verilog is a hardware description language (HDL) for developing electronic devices like embedded systems. This is a widely used language in the field of embedded system programming and offers very low-level access to system hardware. You can access and control almost every hardware-specific detail by incorporating this language into your embedded system development.

Assembler

ASM

Machine code is the most basic code that can be used by the processor unit. The code is normally in the hex code and provides basic instructions for each operation of the processor. This type of code is rarely used for embedded systems these days.

Writing in the machine's code is very laborious and time consuming. It is difficult to understand and to search for errors in the code. To overcome this limitation, high-level programming languages like C, C++ and others are often used. When you want to keep your project as compact as possible, Assembler is the language of choice. Assembler is a way to pack and to build clean machine code that is ultimately done by the processor. The advantage is that an expert can use optimisation tricks that just are not available in any other programming language.

PART 2 THE INTERNET OF THINGS

6 WHAT IS THE INTERNET OF THINGS?

As embedded systems were connected to the Internet on a large scale, what we now call the IoT was born. The IoT simply represents any embedded device that connects to the Internet so that it can communicate with other devices connected to the Internet without people being involved. The IoT consists of hardware devices that are equipped with electronics, software, sensors, actuators, and a network that enables the devices to communicate with each other. We can define the IoT as a network of connected devices where each device has an IP address and an embedded system that allows it to communicate with other devices over the Internet.

The IoT is a gigantic network of connected things and people. It includes a huge number of objects of all shapes and sizes: from smart microwave ovens that automatically cook at the right time, to self-driving cars that have sensors that detect obstructions in front of them, to wearable exercise devices that measure your heart rate and the number of steps you have taken that day and then using this information to suggest training plans tailored to you.

A thing in the IoT can be something as different as a ceiling light, a human with a heart monitor, a refrigerator that alerts when goods run out, a jet engine filled with thousands of sensors that collect and send data or a car that has built-in sensors to alert the driver when other cars or objects come too close.

IoT devices are divided into three groups: consumer, enterprise and industrial. Consumer IoT devices include smart TVs, smart speakers, toys, wearables, and smart appliances. Industrial IoT devices include commercial security systems, smart city technologies, smart meters or any device used to monitor traffic or weather conditions. Finally, enterprise IoT devices include smart lighting, smart thermostats and sensors used by companies.

The development of wireless technologies, microelectromechanical systems and the Internet has made the IoT possible. Thanks to inexpensive processors and wireless networks, it is possible to connect everything from shoes to airplanes to the IoT. This allows devices that otherwise would be unintentional, get digital intelligence so they can communicate with each other without people being involved. Almost all possible physical devices can be connected to the Internet. The IoT bridges the gap between the physical and digital world to improve the quality and productivity of people, communities, and industries.

How the Internet of Things is Built

The main components of the IoT are:

- The thing,
- A local network,
- The Internet and
- The cloud.

The thing contains an embedded system that transmits and receives information over a network for the purpose of controlling another device or communicating with a user.

A thing in the IoT has some of the following components:

- A unique Internet address connection,
- A communication device that can send and receive messages,
- Built-in computer software that can perform information processing,
- One or more sensors and/or
- An actuator that can perform actions on the physical environment.

For something to become a thing in the IoT, it must have some or all the features mentioned above. This means that a chair, a refrigerator, or a lamp must contain an embedded system to become a thing in the IoT.

A key component of a thing in the IoT is a microcontroller or microprocessor that can execute software instructions. Another key component is the IPv6 protocol that has a central role in managing all the things that are and will be connected to the Internet. It is estimated that by 2030, 50 billion things will be connected to the Internet (Statista).

Machine-to-Machine Communication

The IoT is based on machine-to-machine (M2M) communication, which refers to direct communication between physical devices using any means of communication, including wired and wireless. M2M communication and the IoT both deal with machines that communicate with each other, but there is a small difference between M2M and IoT. M2M has traditionally been communication between two specific machines. IoT, on the other hand, is equipment that communicates using an IP network and can thus communicate with any embedded device and computer connected to the Internet.

Common applications for M2M have been the traffic control system, telemedicine, company security and telemetry. M2M communication has been used in traffic control. In a typical traffic control system, there are sensors used to monitor the speed and size of traffic. This information is sent to computers that control the traffic. With such incoming data, M2M communication can regulate traffic flow.

6.1 EXAMPLES OF THE INTERNET OF THINGS

The IoT is used in many contexts, and its use has only increased. IoT technology is used in smart homes, smart cities, connected cars, connected portable assets, and associated health care.

Market	Use
Smart Home	Heating, ventilation, air conditioning, lighting control, smart door locks, etc.
Smart Cities	Parking, health, pollution, traffic jams, outdoor lighting, air quality, waste management, fire detection, etc.
Smart Farming	An increasing number of farmers are using IoT-enabled tools to monitor weather, soil composition, soil moisture levels, crop health and growth and livestock activity.
Activity Trackers	These sensor devices are designed to be worn during the day to monitor and to transmit key health indicators in real time, such as fatigue, appetite, physical movement, oxygen levels, blood pressure, fall detection and compliance with taking medicine.
Self-Healing Machines	Relying on arrays of thousands of sensors, artificial intelligence and machine learning, manufacturing equipment can be designed to recognise variances in its own operation and correct them before they turn into problems that require downtime and repair.
Energy Connection	Discover power outages faster
IoT in Health Care	Provide information about people's health
Companion Robots	Companion robots have become a welcome friend. They have names, some can converse and one day they may even be able to administer medicine to elderly people living alone.

Figure 6.1 Examples of the use of the Internet of Things (IoT)

Smart Homes

It would be nice if you could turn on the air conditioning before you get home or if a light could turn itself off after you have left a room or if you could unlock the door of your apartment to others for temporary access, even when you are not at home. The IoT can make life easier and more convenient for people.

Smart Homes are currently popular. Embedded systems can be used extensively in the home and it is in the home that most people probably encounter things connected to the Internet. This is a use of the IoT where the major technology companies compete hard, especially Amazon, Google, and Apple. Smart homes have become a success and it is predicted that smart homes soon will be as common as mobile phones. The cost of owning a house can be a huge expense. The use of IoT technology in the home can save money, time, and energy (Figure 6.2).

Example	Use
Light Control System	You can turn on/off lights in your apartment from a mobile phone.
Heating	Smart thermostats can reduce monthly energy consumption by up to 30%.
Check That the Oven Is Switched Off	Smart outlet can turn on/off any plugged device in your living room via the Internet.
Air Quality	You can monitor the air quality of the home and the level of pollution in the city.
Monitor an Elderly Family Member	Wireless sensors are placed around the home so you can follow a person's daily routine.
Monitor a Baby	Parents receive information about a baby's breath, skin temperature, body position and activity level on the smartphone. This approach could avoid sudden infant death syndrome.
Keep Your Plants Alive	Helps save time and resources by watering and grooming plants, based on their actual growing needs and conditions.

Figure 6.2 Some examples of the use of the Internet of Things in homes

Smart homes use automated and intelligent features that contribute to lower energy consumption, better comfort, easier operation, and a higher level of safety. Smart homes often use different control systems for light, heat, the refrigerator, fire protection, burglary protection and ventilation. From the mobile phone, you can control light, heat, power consumption, blinds, the garage door, the exterior door, cameras, ventilation, sound, and pictures in your home.

Smart homes can help older people live longer in their own homes instead of having to move to retirement homes. IoT equipment makes it easier for family and caregivers to communicate with them and monitor them.

Using the IoT as an aid provides a better understanding of how our homes work and it allows you to save energy, for example, by cutting down on heating costs.

Wearables

Wearables have become popular. Wearables in the form of activity gauges, sports watches, and smartwatches (Figure 6.3) make it easy to track your health, chart how active you are and set personal goals for your physical exercise.

Figure 6.3 Smartwatches have been designed to integrate themselves into every moment of the wearer's life, whether awake or asleep. They record heartbeats, sleep patterns and workouts, among other aspects. The image is a sixth-generation Apple Watch

Wearables are installed with sensors and software that collect data and information about the user. The data are later pre-processed to extract important information about the user.

Wearables can be used to collect data on a user's health, including:

- Heart rate,
- Blood pressure,
- Calories burned,
- Time spent exercising,

- Steps walked,
- Seizures,
- Physical strain and
- Release of certain biochemicals.

With an action camera mounted on your body, it is easy to film what you see wherever you are and what you experience. The prerequisite for such IoT technology to provide usable application is that they are energy efficient and have a small size. Wearables have experienced an explosive demand worldwide. Companies like Apple, Google and Samsung have invested heavily in building such devices.

Smart Cities

A smart city is an application of the IoT that is very interesting. Smart monitoring, automated transport, smarter energy management systems, water distribution, city security and environmental monitoring are all examples of the IoT used in smart cities (Figure 6.4). By spreading many sensors over a city, the authorities will get a better idea of what is happening in real time. As a result, smart city projects are a key element of the IoT. Cities already generate large amounts of data from security cameras and environmental sensors and already contain large infrastructure networks used for control, such as traffic lights.

Smart Home	Smart Buildings	Smart Energy
Smart Parking	Smart Environment	Smart Street Lights
Traffic Management	Waste Management	Air Pollution
Public Safety	Intelligent Shopping	Electric Vehicle Charging

Figure 6.4 Some uses of the Internet of Things in smart cities

The IoT will solve major problems for the population of cities such as pollution, traffic congestion, lack of energy supply and so on. Sensors with mobile communication allow you to send alerts to municipal services when a trashcan can be emptied. By installing sensors and using web applications, citizens could find available parking spaces throughout the city. The sensors could also detect general errors and any problems in the city system.

Smart city spans several applications, from environmental monitoring to water distribution, waste management, traffic management and city security. Its popularity is driven by the fact that many smart city solutions promise to reduce the problems for people living in cities. IoT solutions in smart cities reduce noise and pollution, solve traffic congestion problems, and help make cities safer.

Sensors could help the elderly in their daily life, while others could keep track of whether a beach has become too crowded and then offer swimmers another option. Other examples are monitoring infrastructure such as roads, bridges, and railways with sensors to investigate structural changes such as cracks and tiles (Figure 6.5). The ability to understand better how a city works should enable governments to make changes and monitor how this improves citizens' lives.

Example	Use
Smart Parking	Monitor available parking in a city
Structural Health	Monitor vibrations and material fatigue in buildings, bridges, etc.
Noise Pollution	Monitor noise in urban areas
Smartphone Registration	Register equipment that uses Wi-Fi or Bluetooth
Electromagnetic Measurement	Measure radiation in homes or urban areas
Traffic Jams	Monitor vehicles and pedestrians on the roads
Smart Lights	Intelligent streetlights that adapt to the day and night as well as to weather
Garbage Treatment	Keep track of the amount of garbage in containers
Smart Roads	Intelligent highways that warn about weather and traffic conditions

Figure 6.5 Some examples of the Internet of Things in smart cities

Industrial Internet of Things

The Industrial Internet of Things (IIoT) is a new trend in the industrial sector. It comprises industrial engineering with sensors, software, and great data analysis to create intelligent machines. The IIoT could have many of the same uses and benefits as the IoT. You could integrate smart sensors into manufacturing machinery, energy systems and infrastructure such as piping and wiring. These sensors, through the data they collect and the advanced functionality they enable, could help industrial businesses to boost their efficiency, productivity, employee safety and more.

The idea behind the IIoT is that smart machines are more accurate in processing data than humans. Moreover, these data can help companies detect inefficiencies and issues earlier. The IIoT has great potential for quality control and data processing. Opportunities for stock information, tracking of goods and automated delivery would increase the efficiency of the supply chain.

Here are some IIoT use cases and impacts:

- Predict machine maintenance,
- Help minimise downtime in factories,
- Increase throughput by considering real-time demand,
- Energy savings,
- Safety systems such as thermal sensing, pressure sensing and gas leaks and
- Factory floor expert systems.

Connected Cars

A connected car is equipped with Internet access and usually also with a wireless local area network (LAN). This allows the car to communicate with the driver, other cars, roadside infrastructure, or the cloud. This technology is not only able to improve traffic safety, but also to improve efficiency and comfort.

The car's digital technology has focused on optimising the vehicle's internal functions. Now the focus is on improving the car experience. A connected car can optimise its own operation and maintenance and improve passenger comfort with the help of indoor sensors. Most large car manufacturers work with connected car solutions.

The IoT prevents accidents and improves vehicle safety. A vehicle can itself continuously monitor the near traffic and intervene if the driver is inattentive. Thanks to IoT connectivity, a car's many software-reliant components can be updated via the Internet.

The Internet of Things in Agriculture

With the continuous increase in the world's population, the demand for food has increased markedly. Governments help farmers by using advanced techniques and research to increase food production. Smart farming is one of the fastest growing fields in the IoT. Farmers use information from IoT data to provide a better return on investment. For outdoor farming, the IoT can measure soil moisture and pay attention to the weather, so that smart irrigation systems only water when needed and thereby reduce water consumption. For indoor agriculture, the IoT offers the possibility of monitoring and controlling climate conditions such as humidity, temperature, light, etc. This ability should increase production.

The Internet of Things in Rearing Livestock

When rearing livestock, animal husbandry and cost savings are crucial. By using IoT equipment to collect health and well-being data for cattle, farmers can become aware of sick animals early, information that could help prevent many diseases. Due to the size of the farming business and the large number of livestock that can be monitored, the IoT could revolutionise the way farmers work. Smart farming has become an important area of application in IoT for countries that predominantly export agricultural products.

Smart Sales

The potential of the IoT in the sale of goods is enormous. The IoT gives retailers an opportunity to connect with customers to improve their experience in the store.

Smartphones represent a way for retailers to remain connected to their consumers, even outside the store. Interaction via mobile phones could help resellers provide their consumers with better services. They could use IoT equipment to track how customers move through a store, improve the store environment and place important goods in more trafficked areas. They could use available sales data to identify which items are selling fastest and automatically adjust the sales data with supply so that popular items do not run out of stock. The information provided by connected devices would enable retailers to make smart decisions about which goods to fill, helping to save time and money.

Energy Transfer

The power grid of the future will not only be smart, it will also be very reliable. The basic idea behind smart grids is to collect data automatically and analyse the behaviour of consumers and power suppliers to improve the efficiency and economy of electricity. You would also be able to detect causes of power outages faster.

Energy Efficiency

People and organisations can achieve significant reductions in energy consumption by using the IoT. Sensors monitor lighting, temperature, and energy consumption, among many other factors. The data are processed by software to evaluate real-time activities. For example, smart thermostats can automatically turn off heat/cooling to save energy when nobody is home.

The Internet of Things in Health Care

The IoT has great potential application in health services. The concept of connected health care and smart medical equipment has enormous potential. Bringing the IoT into medicine would help provide stronger, healthier, and easier patient care. From the implantation of medical devices to smart sensors, the IoT would speed up health care delivery, enabling doctors to spend less time on transportation, diagnose illnesses and communicate with patients.

The IoT allows for increased monitoring and recognition, which could help improve health and make life safer for people. Real-time monitoring via connected devices could save lives in the event of a medical emergency like heart failure, diabetes, asthma attacks, etc.

Researchers have shown that the IoT in health services will be massive in the coming years. The IoT in the health care system aims to allow people to live a healthier life by using connected devices. The collected data should enable personal analysis of a person's health and provide tailored strategies to combat diseases.

Disaster Warning

Sensors can collect critical information about the environment, and this information enables early detection of environmental disasters such as earthquakes, fires, tsunamis, etc. This endeavour would save human lives.

Law Enforcement

Better monitoring and investigation would allow the authorities to detect when a criminal act has occurred and respond much more quickly so that citizens can live more safely. Better law enforcement would even be able to predict crime and stop it before it happens.

Elderly

Monitoring the elderly could be lifesaving: it could detect automatically when someone falls or when someone has a heart attack, so help could be sent immediately.

Environmental Quality

Sensors can also detect radiation, pathogenic agents, and poor air quality so that hazardous concentrations can be identified early, and people evacuated.

Smart Environment

The idea behind smart environments is to build an environment with embedded sensors and computing devices to better understand and control the environment (Figure 6.6).

Example	Use
Discovery of Fires	Monitor forest areas
Air Pollution	Control carbon dioxide in factories and towns
Earthquake	Monitor earthquake-exposed areas
Water Quality	Monitor the quality of drinking water
Swimming Pools	Check the water quality in swimming pools
Pollution in the Sea	Monitor the level of pollution in the sea
Water Leakage	Monitor humidity outside of water containers and water pipes.
Floods	Monitor water levels in rivers, ponds, and reservoirs
Radiation Levels	Monitor radiation at nuclear power plants
Gas Leakage	Monitor gas levels and leakage in industry
Indoor Air Quality	Monitor toxic gases and oxygen levels
Temperature	Control temperature in refrigeration disks in industry and medicine
Weather	Monitor weather conditions
Keep the Streets Clean	Use real-time data collection and alert when municipal services should know that a trash can needs to be emptied
More Efficient Street Lighting	Smart lighting systems allow a city to provide the right level of lighting regardless of the time of day, season, and weather conditions.

Figure 6.6 Some examples of the Internet of Things in smart environments

6.2 ADVANTAGES AND DISADVANTAGES OF THE INTERNET OF THINGS

The IoT is beneficial for businesses because it enables the collection and analysis of data from production equipment, weather stations, smart meters, automobiles, and other types of machines. IoT analysis programs help companies understand data from sensors, with a view to reducing maintenance costs, avoiding equipment failures, and improving operation. In addition, consumer goods retailers, restaurants and manufacturers use data from smartphones, portable technologies, and home appliances to make marketing and campaigns more targeted.

To understand what impact the IoT can have on people's way of life, it is important to review its advantages (Figure 6.7) and disadvantages (Figure 6.8).

Advantage	
Control	You can control a device that is miles away in real time.
Automation	Automation enables performing tasks without human intervention. Automating tasks in a business helps increase the quality of services and reduces the need for human impact.
Information	Accessing information is easy: a person can access information from anywhere in the world. The more information, the easier it is for people to make good decisions.
Monitoring	The second most obvious benefit is that monitoring sensors provide information about the local environment – for example, to know when a refrigerator is empty or how good the air quality is in a house.
Economy	The economic aspect is the best advantage. The technology can replace people responsible for monitoring and maintenance. Optimal utilisation of energy and resources can be achieved by using IoT technology and keeping the devices under surveillance. The IoT can alert you to any bottlenecks, breakdowns, and damage to the system. You will therefore save money by using IoT technology.
Save Time and Greater Efficiency	The IoT helps people perform daily work tasks. This saves valuable time. Instead of performing monotonous tasks every day, it is possible for people to do other more creative jobs. The time savings due to the IoT can be quite large.
Better Quality of Life	All uses of IoT technology result in increased comfort, convenience, and better control, with an overall improvement in the quality of life for humans.

Figure 6.7 Some advantages of the Internet of Things (IoT)

Disadvantage	
Complexity	The IoT is a large and complex network that connects a huge number of different devices. A small error can affect the entire system, and the more complex the systems are, the greater the possibility of errors.
Less Work for People	When tasks become automated, there will be less need for human labour; this aspect will affect employment. In a future with the IoT, there will be a decline in the hiring of low-skilled employees.
Security	All IoT devices that a person uses are connected to the Internet. Hence, there is a great risk that personal data could be leaked. It is a major disadvantage of sharing information: confidential information might not be safe and could simply be hacked by unauthorised people.
Privacy	Privacy is a major issue with the IoT. All data must be encrypted so that information about your financial status or when there is no one home in your apartment is not available to other people or criminals.
Addiction	Today's lifestyle is technology driven. We depend on technology for the smallest of tasks.

Figure 6.8 Some disadvantages of the Internet of Things (IoT)

The IoT makes our homes, offices, and vehicles smarter, more measurable, and communicable. Although the IoT has some drawbacks, the benefits of saving consumers' time and money cannot be ignored. We must find ways to combat the disadvantages.

Security systems at home make it easier to monitor what is going on, or to see and to talk to visitors. Meanwhile, smart thermostats can help us warm up our homes before we get there, and smart light bulbs can make it look like we are at home even when we are out.

Sensors can also help us understand how noisy or polluted our environment can be. Autonomous cars and smart cities can change how we design and treat our local environment.

7 HOW THE CLOUD WORKS

7.1 WHAT IS THE CLOUD?

The cloud consists of remote servers that can be used via the Internet. Using the cloud involves storing and processing data on remote servers rather than on local and privately owned computers (Figure 7.1).

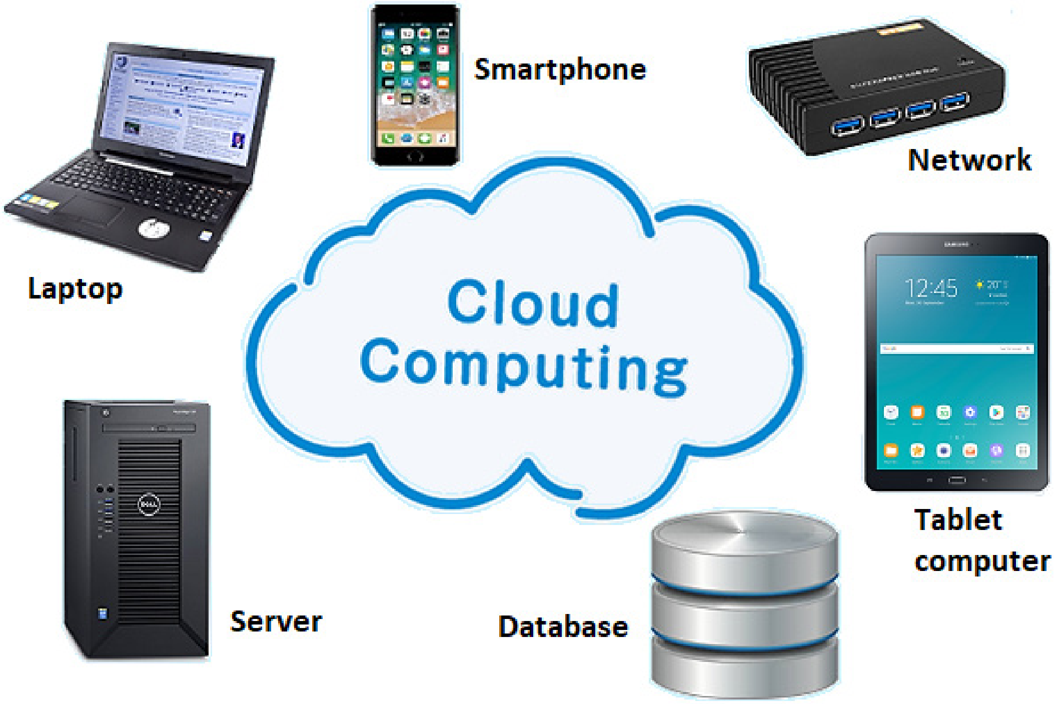


Figure 7.1 Cloud computing

The cloud is the supply of data services over the Internet, including servers, data storage, databases, networks, software, analysis and more.

You can store small or large files in the cloud, and you can get hold of them everywhere via the Internet. You can use databases in the cloud. Computers in the cloud can process small or large amounts of data for you at any time.

Here are some usual usages of the cloud:

- Save, backup and restore data;
- Have websites and blogs;
- Create new apps and services;
- Stream audio and video;
- Deliver software on request; and
- Analyse data to draw conclusions.

You probably use the cloud even if you are not aware of it. If you use an online service to send email, edit documents, watch movies or TV, listen to music, play games, or save photos and files, it is likely that the cloud makes this action possible.

Companies offering these computing services are called cloud providers, and they usually charge for their cloud services based on how large the usage is, just as you pay for electricity you use in your home. You usually pay only for the cloud services you use, and this may reduce your operating costs. This makes the infrastructure more efficient, and the services are scaled as the company's needs change. The first cloud services came around 2006, and soon after several organisations started using this technology for various reasons.

The huge amount of data that IoT applications generate has meant that many companies choose to use cloud computing instead of buying and using their own servers. The cloud giants already dominate these companies. Microsoft has its Azure IoT package, Amazon Web Services offers a variety of IoT services and Google has Google Cloud.

Features of the Cloud

According to the National Institute of Standards and Technology (NIST), these are some specific features that define the cloud:

- Self-service on demand,
- Wide network access,
- Fast elasticity or expansion,
- Pay as you go and
- Measured service.

Cloud computing resources can be provisioned without human interaction from the service provider. In other words, a cloud user can provision additional computing resources as needed without contact with the cloud service provider. This can be a storage space, virtual machine instances, database instances and so on.

Cloud computing resources are available over the network and can be accessed by diverse customer platforms. In other words, cloud services are available over a network, ideally high broadband communication link such as the Internet, or in the case of a private clouds it could be a LAN.

One of the great things about cloud computing is the ability to provision resources quickly in the cloud as businesses need them, and then to remove them when they are no longer needed. Cloud computing resources can scale up or down rapidly and, in some cases, automatically, in response to business demands. It is a key feature of cloud computing. The usage, capacity and, therefore, cost can be scaled up or down with no additional contract or penalties.

In cloud computing, the user must pay only for the service or the space they have utilised. The service is economical and most of the time some space is allotted for free. However, be aware that prices suddenly may rise in the future.

The Benefits of the Cloud

Without the cloud, the IT growth and market would be non-existent. Essentially, billions of endpoint devices that were historically dumb and not connected would need to manage themselves without the ability to share or aggregate data. The cloud provides the ability to have simple sensors, cameras, switches, beacons, and actuators participate in a common language with each other. The cloud is the common denominator of the data currency.

The cloud is a big change from the traditional way companies have used IT resources. We will look at why the cloud is so popular (Figure 7.2).

Benefits	
Reduced Cost	The cloud eliminates the cost of buying hardware and software and setting up and running local data centres. It saves space and electricity around the clock for power and cooling, and you do not need your own information technology (IT) experts to manage the infrastructure. The cloud saves businesses space, work, and money.
Saves Time	Most cloud services are self-service and provided on demand, so even large amounts of computing resources can be delivered in minutes, often with just a few mouse clicks. This gives companies great flexibility and removes the pressure on capacity planning.
Scaling	A major advantage of cloud services includes the scaling capability. In the cloud language, it means delivering the right amount of IT resources at the right time – for example, computing power, storage and proper bandwidth as needed.
Productivity	Local data centres usually require a lot of equipment, hardware setup, software uploads and various time-consuming IT management efforts. The cloud eliminates the need for many of these tasks, so IT teams can concentrate on other important tasks.

Benefits	
Performance	The largest cloud computing services run on a worldwide network of secure data centres that are regularly upgraded to the latest generation of fast and efficient hardware. This provides better performance than a local business data centre.
Reliability	The cloud makes data backup and disaster recovery simpler and more affordable. Data are stored more securely as copies can be located on servers in various locations around the world. If a server burns or a data centre drops out of operation, one can obtain the data from another data centre.

Figure 7.2 Common reasons why organisations use cloud services

Disadvantages of the Cloud

There are also some disadvantages of using the cloud that should be taken into consideration (Figure 7.3).

Disadvantages	
Network Connection Dependency	You need a network to send files to the cloud and retrieve them. If you lose your network connection because of a storm or an outage, you may experience some downtime.
Limited Features	Not all cloud providers are created equally. When you use cloud computing for storage and backup, you should ideally be working with a provider that offers the value of unlimited bandwidth. You may also experience limited storage space or accessibility.
Loss of Control	You, essentially, trust another party to take care of your data. You are trusting that they will maintain their data centres and servers with the same care as you would, if not more. You must trust that your provider's data centres are compliant and secured both physically and online. Some find the lack of in-house control of the server unnerving.
Security	Cloud hacking cases have shown that not all cloud providers are as secure as they claim to be. As a business, you cannot afford to have sensitive information about your company, or your clients fall victim to hackers. One of cloud computing's greatest disadvantages is that you do not always know which providers you can trust.
Technical Issues	If you experience any technical issues, you have no choice but to call your hosted provider's technical support for help. You cannot fix your cloud computing problems in-house, and some providers do not offer around-the-clock technical support.

Figure 7.3 Some disadvantages of cloud services

7.2 THE CLOUD ARCHITECTURE

Cloud computing refers to the components and subcomponents required in the cloud. These components usually consist of:

1. A front-end platform that can be a computer or a mobile device and
2. Back-end platforms that consist of servers, and a network such as the Internet or an Intranet.

Combined, these components constitute the cloud architecture. In a cloud architecture, all applications are controlled, managed, and operated by a server in the cloud. The computer system is copied and preserved remotely as part of the cloud configuration. A good cloud computing system can create virtually unlimited efficiency and opportunities.

Front-End and Back-End

It is useful to divide a cloud-based system into front-end and back-end. Front-end is the part near the user or client. Back-end is the part of the system that is in the cloud. They are connected to each other via a network, usually the Internet.

Front-end is the visible interface that computer users or mobile users use. It includes the client's computer and computer network as well as applications required to access the cloud system. Not all cloud systems have the same user interface. Services such as web-based email programs utilise existing browsers such as Internet Explorer or Firefox. Other systems have different applications that provide network access for clients.

The back-end system includes all the resources required to provide cloud services. A system's back-end could comprise a variety of servers, data storage facilities, virtual machines, a security mechanism and services, all built in accordance with a distribution model, and all are responsible for providing a service. The back-end system is the various computers, servers and data storage systems that are in the cloud. In theory, a cloud computing system could include virtually any computer program imaginable, from computing to video games. Usually, each application would have its own server.

Cloud Client Platforms

Cloud computing architectures consist of front-end platforms called clients or cloud clients. These clients are servers, fat clients, thin clients, tablets, and mobile devices. These client platforms interact with the cloud via intermediate software.

How the Cloud Works

Cloud computing resources are provided by server-based applications via digital networks or via the public Internet. The programs are available to users via mobile and desktop devices. A central server manages the system and monitors traffic and the client to ensure everything runs smoothly. It follows a set of rules called protocols and uses a special type of software called middleware. Middleware allows network-based computers to communicate with each other.

Cloud Storage

Network storage comprises data stored and available to several customers. Cloud storage is usually distributed in the following configurations: public cloud, private cloud, community cloud or a combination of the three also known as a hybrid cloud. To be effective, cloud storage must be flexible, scalable, and secure.

If a cloud firm has many customers, there would be a high demand for storage space. Some companies have hundreds of digital storage devices. Cloud computing systems require at least twice as much storage space as required to keep all customer information stored. That is because these devices sometimes come out of operation, which could happen to any computer. A cloud system must have a copy of all customer information and store it on other devices. The copies allow the central server to access data that would otherwise be lost. Copying data as backup is called redundancy.

Server Virtualisation

It is possible to have several virtual servers on one physical server, with each virtual server running its own independent operating system. This technique is called server virtualisation. By maximising the performance of individual servers, server virtualisation reduces the need for physical machines.

Virtualisation uses multiple virtual machines on a physical computer or server. This approach achieves better scalability and workload while using fewer servers. Hence, less power is used, and money is saved on infrastructure and maintenance.

To organise virtualisation, a special operating system for this called a hypervisor is used (Figure 7.4). A hypervisor uses protocols that allow multiple virtual machines to run simultaneously on a physical server. The hypervisor controls the communication between its containers of virtual machines and the connected world beyond.

Server virtualisation used by hypervisors circumvents some of the physical limitations that stand-alone servers can face. Virtualisation refers to the creation of a virtual machine that acts like a real computer with an operating system. This approach provides better hardware utilisation.

Hypervisor Type	Behaviour
Native	These hypervisors run directly on a single metal server without an intermediate operating system and thus have full responsibility for performance and reliability.
Hosted	These hypervisors run on a conventional operating system just as other computer programs do. This type of hypervisor is beneficial for both private and public cloud computing to achieve performance improvements.

Figure 7.4 Different types of hypervisors

8 HOW THE INTERNET OF THINGS WORKS

The significant value of the IoT is in the interpretation and decision made using the IoT data. The value of the IoT is what the data can tell us. Analysing data from IoT devices underlies the creation of more efficient systems. Smart IoT systems make it possible to automate many tasks. This applies to tasks that repeat themselves or are time-consuming or dangerous.

What we want to achieve with an IoT system is usually:

- Communication,
- Control and
- Cost Savings.

Communication

The task of an IoT device is to collect and to communicate information. For example, a sensor can measure the temperature of a refrigerator and send a message to a cell phone if the temperature becomes too high or too low. Another example is an IoT device reporting whether an air filter is clean and working properly.

Control

You should be able to control an IoT device over the Internet or it should be able to control itself – for example, a lamp that can be switched on and off using a mobile phone app. You can also use the IoT to start your washing machine from your mobile phone and when the washing is finished, the washing machine can send a message to your phone.

Cost Savings

The desire to save money is the most common reason for businesses to adopt the IoT. While many companies will use the IoT to save money, individuals will use the IoT to automate home processes. However, individuals also want to save money sometimes. An example is to use the IoT to reduce heating costs in apartments or reduce fuel costs while driving.

8.1 THE MAIN COMPONENTS OF THE INTERNET OF THINGS

The IoT refers to billions of physical devices around the world that are connected to the Internet. Thanks to cheap processors and wireless networks, it is possible to connect everything from lamps to aircrafts to the IoT. Hence, devices that would otherwise be unintentional get digital intelligence so they can communicate with each other without people being involved.

An IoT system consists of several different components (Figure 8.1). The thing is a central component of an IoT solution. The thing collects information, which is used to get insights. These insights will be used to make decisions that result in actions.

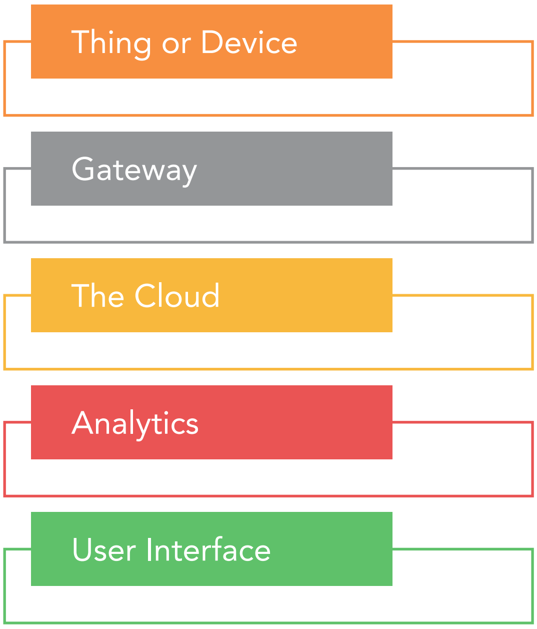


Figure 8.1 Major components of the Internet of Things

The thing is an embedded computing device or an embedded system that transmits and receives information over a network for the purpose of controlling other devices or interacting with a user. The thing always has either a microcontroller or a microprocessor-based device.

A thing can be a car or a washing machine, or it can be larger systems such as a jet engine filled with thousands of sensors that collect and transmit data. Smart city projects can fill entire regions with sensors to help users understand and control the environment.

Almost every physical object can be transformed into an IoT device if you can attach an embedded system to it and connect it to the Internet. However, the concept of the IoT is mainly used for devices that are not normally expected to have an Internet connection but can communicate with the network regardless of human action. For this reason, a PC is not considered an IoT device.

8.2 SOME BASIC TECHNOLOGIES IN THE INTERNET OF THINGS

The IoT will continue to grow as new IoT technologies are developed and introduced. A problem with new technologies is that it requires training to use them. To teach employees to use new technologies will be an important challenge for organisations using the IoT.

There are many technologies related to the IoT and many of them are new. In this section, we shall look at some technologies important for the IoT.

IoT Platforms

The IoT strives to connect devices remotely for seamless functioning and ease of operations. An IoT platform bridges the gap between device sensors and a data network. It provides an insight into the data used in back-end applications. An IoT platform is a set of components that allows developers to spread out the applications, to collect data remotely, to secure connectivity and to execute sensor management.

IoT platforms combine in a single product many of the components of the infrastructure of an IoT system. IoT platforms originated in the form of IoT middleware, the purpose of which was to function as a mediator between the hardware and application layers. For developers, an IoT platform provides a set of ready-to-use features that greatly speed up development of applications for connected devices as well as take care of scalability and cross-device compatibility.

Internet of Things Device (Thing) Management

The things in the IoT require control and monitoring. This includes monitoring devices, software updates, diagnostics, crash analysis and reporting, physical management, and security management. There is a need for tools that can manage and monitor thousands and perhaps even millions of IoT devices.

IoT Analysis

We want to exploit the information collected by IoT devices in many ways, which will require new tools and algorithms for analysis. As data volumes increase over the next few years, analysis of IoT data will be different from traditional analysis.

Low-Power, Short-Range Internet of Things Network

Low-power, short-range networks will dominate wireless IoT connectivity, instead of connections using IoT networks over a wide area. Several different solutions will coexist, without any dominant wireless technology.

Low-Power, Wide-Area Internet of Things Networks

Traditional mobile networks have not provided a good solution for those IoT applications that need wide coverage combined with relatively low bandwidth, good battery life, high connectivity density and low costs. Future standards such as narrowband Internet of Things (NB-IoT) will probably dominate this need. NB-IoT is a low-power, wide-area network (LPWAN) developed by the 3rd Generation Partnership Project (3GPP) to enable a wide range of mobile devices and services. The 3GPP is a standards organisation that develops protocols for mobile telephony.

Internet of Things Processors

The processors and architectures used by IoT devices determine what capabilities they have, such as how good they are in terms of security, encryption, and power consumption, and whether they have been developed sufficiently to support operating systems, up-to-date firmware, and embedded devices. Understanding the consequences of choosing a processor will require good technical knowledge.

Internet of Things Operating Systems

Traditional operating systems such as Windows and iOS were not designed for IoT applications. They use too much power, need fast processors and in some cases lack features that guarantee real-time response. They also use too much memory for small devices, and they may not support the chips that IoT developers use. Therefore, a wide range of operating systems has been developed for the application of IoT devices to suit many different types of hardware and functional needs.

Event Stream Processing

Some IoT applications will generate extremely high data rates that need to be analysed in real time. Systems that create tens of thousands of events per second are common, and millions of events per second can sometimes occur. To meet such requirements, distributed platforms have been developed; they can process high-speed data streams and perform real-time analysis and pattern identification tasks.

IoT Standards

Standards and APIs will be crucial because IoT devices must interact and communicate, and many IoT business models will be based on sharing data between multiple entities and organisations. Many IoT systems will show up, and organisations that make IoT products may need to develop more variants to support different standards and systems and be prepared to update products during their lifetime when standards evolve, and new standards and APIs arrive.

IoT Security

Security technology will be needed to protect IoT devices and platforms from hacking and physical manipulation. IoT security is complicated because many IoT devices use simple processors and scaled-down operating systems that may not support advanced security features.

8.3 A MODEL FOR THE INTERNET OF THINGS

A complete IoT system has five important elements:

- Sensors or units with sensors,
- Actuators,
- Connection to a network,
- Data management and
- A user interface.

Sensors or Units with Sensors

The IoT is often used to process measured data. Getting data from IoT devices anywhere in the world, connected via the Internet, provides the basis for control and optimisation. This is what forms the basis for smart city and smart grid design.

First, sensors collect data from the environment in which it is located. This can be as simple as a temperature reading or something more complicated like a video recording. The expression units with sensors are used here because a device can have several sensors. However, if the thing is a stand-alone sensor or a unit with sensors, in this first step data from the local environment is captured.

Sensors are a key element in the connected networks for the IoT that are increasingly being used in smart city and smart grid design. Connecting sensors to the IoT is relatively easy in some cases, but in other contexts requires more consideration. The data are sent as data packets to the IoT, usually using the IPv6 protocol. Digital interfaces for the latest sensors simplify the IoT interface to improve system reliability and functionality.

Connection

Data from sensors is sent to the cloud, but they need a way to get there. Sensors and devices can be connected to the cloud using a variety of methods, such as mobile, satellite, Wi-Fi, LPWAN or connecting directly to the Internet via ethernet. The various options have advantages and disadvantages in terms of power consumption, range, and bandwidth. Selecting which connection option is best depends on the IoT application, but all the connection options perform the same task: getting data to the cloud.

Data management

When the data reach the cloud, software performs some form of processing on it. This can be very easy – for example, to check that the temperature reading is within an acceptable range. On the other hand, it can also be more complicated, such as using video software to identify objects such as an intruder in a house. A user should decide what happens if the temperature is too high or if there is an intruder in a house. Therefore, there is need of a system that sends information to the user.

User Interface (Mobile Applications)

Mobile applications (also known as mobile apps) are software programs developed for mobile devices such as smartphones and tablets. The IoT is a network of Internet-enabled devices all having an IP address and communicating with a user through a mobile app on a smartphone interface.

The information from IoT devices should be presented to an end-user in one way or another. This can be via a notification to the user via e-mail or Short Message Service (SMS). For example, a text alert can be sent when the temperature in a refrigerator is too high. In addition, a user can have an interface that allows them to examine the system: they can check the video footage in the house via a phone app or a web browser. A user often can interact with the system. For example, the user could remotely adjust the temperature of a refrigerator via an app on a mobile phone. In addition, some actions could be performed automatically. Instead of waiting for a user to adjust the temperature, the system could do so automatically using a program. Moreover, instead of calling the user to warn about an intruder, the IoT system could automatically alert the police.

Summary

An IoT system consists of sensors that communicate with the cloud through some form of a connection. When the data reach the cloud, the software processes it and may then decide to perform an action, such as sending an alert or automatically adjusting the system without notifying the user. However, if user access is required or if the user wants to check the system, a user interface could be used. Any customisations or actions the user makes are then sent in the opposite direction through the system. A message from the user interface is sent to the cloud and then back to the sensor unit to make a change.

9 BIG DATA AND THE INTERNET OF THINGS

The IoT can often produce huge amounts of data. Businesses can analyse data to gain knowledge of how components behave in real-life situations. This endeavour could help companies make improvements much faster. For example, data generated from sensors around a city could help city planners make the city community more efficient.

In the future, the IoT will generate larger and larger amounts of data. Hence, companies would need to upgrade their current storage systems, tools, and technology to be able to handle huge amounts of data and to take advantage of the insights that Big Data can provide.

9.1 BIG DATA

Big Data comprises datasets that are very large and complex. Therefore, special software is used; it has been designed to process large datasets. Large data tasks include data collection, data storage, data analysis, searching, sharing, transferring, visualisation, updating, querying, information, and privacy.

Big Data often refers to the use of predictive analysis. Advanced data analysis methods and machine learning are used to extract information from data. The analysis of the datasets can find relationships that can show business trends, prevent diseases, fight crime, and so on.

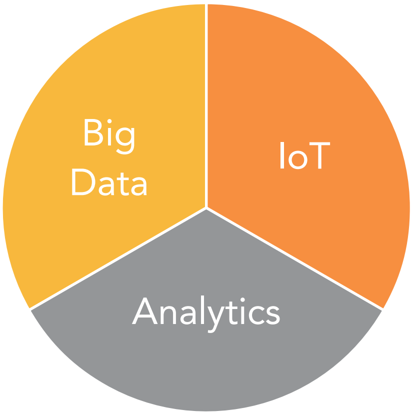


Figure 9.1 Big Data and analytics are closely connected to the Internet of Things

The datasets grow continuously because they are increasingly being assembled by many inexpensive sensors on the IoT, and they are transmitted via mobile devices, antennae, microphones, cameras, radio frequency identification (RFID) readers and wireless sensor networks.

Relational database systems, stationary statistics, and software packages to visualise data often have difficulty managing large amounts of data. The work may require massively parallel software running on a variety of servers. What counts as large amounts of data can vary depending on the user's capabilities and their tools. When organisations face hundreds of gigabytes of data for the first time, they might need to rethink data management options.

Data Storage

A consequence of the IoT is an increase in the volume of data that comes to the data storage system of businesses. There is a need to create new data centres to handle all this data.

Due to the enormous burden that IoT data will have on storage systems, organisations have begun to use a cloud-based solution, rather than using their own storage infrastructure. Unlike internal computer systems that need to be updated continuously as the data load increases, cloud storage provides flexibility, scalability, regulatory compliance, and a suitable architecture for storing the IoT data. Cloud storage options include public, private and hybrid models. If a company has sensitive data that is subject to regulatory compliance requirements that necessitate increased security, it would be best to use a private cloud. For other companies, a public or hybrid cloud could be used to store IoT data.

Businesses Need Big Data Technologies

Most companies need to improve their technologies to handle the vast amount of IoT data that will come. The most important task is to be able to receive data from IoT-linked devices. Devices can be connected to each other via Wi-Fi, Bluetooth, or other technology, and they must send required messages using a well-defined protocol. One of the most used protocols is MQTT, which is short for Message Queue Telemetry Transport. MQTT is a lightweight messaging protocol for small sensors and mobile devices, optimised for high-latency or unreliable networks. It is designed for connections with remote locations.

Once data are received, the next step is to find the best technology platform for storing IoT data. Many companies use Spectrum Scale, Hadoop Distributed File System (HDFS) or Lustre FS to store data. However, non-Structured Query Language (NoSQL) databases such as Apache CouchDB are more suitable for IoT data because they provide low latency and high throughput.

9.2 DATA COLLECTION

Data collection is the process of measuring values in the physical world and transforming these quantities into digital values that can be processed by a computer. Data collection systems, abbreviated as DAS or DAQ, convert analogue waveforms into digital values for processing. The data acquisition system components include sensors for converting physical sizes to electrical signals, and analogue-to-digital converters, to convert analogue sensor signals to digital values.

There are also open-source software packages that provide all the necessary tools to retrieve data from IoT devices. These tools come from scientific environments where complex experiments require fast, flexible, and customisable software.

9.3 DATA AGGREGATION

Data aggregation is any process in which information is gathered and expressed in a summary form, for purposes such as statistical analysis.

A common aggregation goal can be to get more information about specific groups of people based on specific variables such as occupation, age, or income. The information about such groups could then be used to customise websites with specific content and advertising that could appeal to a person belonging to one or more groups for which data are collected. For example, a site that sells music CDs might advertise certain CDs based on the age of the user and the data aggregate for their age group.

9.4 MACHINE LEARNING IN THE INTERNET OF THINGS

Machine learning is a subfield of computer science and is a type of AI that provides machines with the ability to learn without explicit programming. Machine learning has evolved from pattern recognition and computational learning theory. In a very basic sense, machine learning in technology today is the process of elimination of human intervention wherever possible. It allows the data to learn patterns by itself and make autonomous decisions without a human having to write new code.

Devices connected to the IoT will create huge amounts of data, all of which will be collected and stored. This will be put into useable formats and silos by Big Data. Machine learning will then use these huge oceans of data to improve processes and to increase self-sufficiency of systems. These processes are then fed back into the devices connected to the IoT and the process can start again.

9.5 THE ROLE OF DATA ANALYSIS IN THE INTERNET OF THINGS

The value of an IoT system is not that it is a single-sensor event or million-sensor events archived away. A significant value of the IoT is in the interpretation and decision made by using the IoT data. The value of the IoT is what the data can tell us.

Data analytics is defined as a process used to examine large and small datasets. The difference between data analysis and data analytics is that the latter is a broader term. Indeed, data analytics also includes tools and techniques that data analysis does not.

The purpose of data analytics is to draw meaningful conclusions from datasets. These conclusions are usually in the form of trends, patterns and statistics that assist business organisations in effective decision-making. Data analytics plays an important role in the growth and success of IoT applications and investments. The analytics tools allow the business units to utilise their datasets effectively.

The key concepts in data analysis are volume and structure.

Volume

IoT applications use large amounts of data. Corporate organisations must handle and analyse these large amounts of data. These datasets along with real-time data can be analysed easily and efficiently with data analysis programs.

Structure

Datasets from IoT applications can be characterised as unstructured, semi-structured and structured. There can be a significant difference in the data formats and data types. Data analysis will allow analysing all these varying sets of data using automated tools and software.

Types of Data Analysis

There are various types of data analysis that can be used on data from the IoT for information. Some of these types are streaming analytics, spatial analytics, time series analysis and prescriptive analysis.

Streaming Analytics

This kind of data analysis processes and analyses large datasets, which is in motion. Real-time data streams are analysed in this process to detect problems and to provide immediate actions. By using streaming analytics platforms, organisations can extract business value from data in motion just like traditional analytics tools would allow them to do with data at rest.

Spatial Analysis

Spatial analysis uses techniques to manipulate, to extract, to locate and to analyse geographic data. This process is used to create geographic models and data visualisations for more accurate modelling and predictions of trend. Location-based IoT applications, such as smart parking programs, can benefit from this kind of data analysis.

Time Series Analysis

Time series analysis could be useful to see how a given asset, security, or economic variable changes over time. There are two main goals of time series analysis: identifying the nature of the phenomenon represented by the sequence of observations, and forecasting (predicting future values of the time series variable). IoT applications, such as weather forecasting programs and health monitoring systems, may benefit from this method of data analysis.

Prescriptive Analytics

Prescriptive analytics is a process whereby data are analysed to provide instant recommendations on how to optimise business practices to suit multiple predicted outcomes. This form of data analysis is used to understand the best practices that can be taken in a particular situation. Commercial IoT applications can take advantage of this kind of data analysis to get better conclusions.

Use of Data Analysis in the Internet of Things

There have been scenarios where IoT investments benefit greatly from the use of data analyses. With change and advancement in technology, there are emerging areas where data analysis can be used in conjunction with the IoT. IoT analyses will also allow increased security and monitoring capabilities through video sensors and the use of data analysis methods.

Health is one of the most important sectors in all countries, and the use of data analysis in IoT-based health programmes can make breakthroughs in this area. The reduction in health care costs, improvement in health care monitoring and remote health care, increased diagnosis and treatment can be achieved by using the same.

The utilisation of data analyses should therefore be promoted within the IoT area to obtain better revenue, competitive profits, and customer engagement.

10 THE ARCHITECTURE OF THE INTERNET OF THINGS

The architecture of the IoT is a framework that defines the physical components, the functional organisation and configuration of the network, operational procedures, and the data formats to be used. However, there is no single standard reference architecture for the IoT because it encompasses a variety of technologies. The IoT architecture can vary significantly depending on the implementation; it needs to be open enough with open protocols so that it can support multiple network applications.

10.1 IMPORTANT TECHNOLOGIES IN THE INTERNET OF THINGS

The IoT represents the nexus of several technologies. We shall look at the following important technologies:

- Sensors and sensor technology,
- IoT gateways,
- Cloud and server infrastructure and Big Data,
- Mobile apps for end users and
- IPv6 addresses.

Sensors and Sensor Technology

A thing is a device equipped with one or more sensors that collect data. The data are most often transmitted via a network. A thing may have one or more actuators that allows action, such as turning on or off a washing machine, or starting or stopping a motor.

Sensors can provide many types of information, ranging from weather conditions and environmental conditions to motion on a conveyor belt, the health condition of a patient or the maintenance data for a jet engine. Sensors are everywhere and capture data from the environment – for example, a temperature sensor that records the temperature in a room and transmits this information further via an IoT gateway.

Internet of Things Gateways

An IoT gateway is an entry to the Internet for an IoT device. Gateways help bridge the local network of sensor units to the Internet or the World Wide Web. They do this by collecting the data from sensor units and transferring it to the Internet infrastructure.

An IoT gateway is the outward connection for an IoT device. Data goes from the things to the cloud and back again through the gateway. A gateway that connects a thing to the cloud allows you to pre-process and filter data before moving them to the cloud. This ability reduces the volume of data sent to the cloud for processing and storage.

IoT gateways also receive commands coming from the cloud to the things. The things can then perform these commands using actuators.

Cloud Computing and Data Analysis

The data transmitted through a gateway is securely stored and processed in the cloud using a Big Data analysis engine. The processed data make our IoT devices smart with the ability to perform actions.

Data analysts can use the data on large storage servers in the cloud to look for patterns and gain insight. Data are analysed and, in many cases, visualised with graphs or diagrams. For example, large data volumes can show the performance of devices, help identify inefficiencies or create a way to improve an IoT system – that is, make systems more reliable and more customer oriented.

Mobile Phone Apps for Users

Mobile phone apps help users to control and to monitor IoT devices. These apps provide information to a mobile phone and let the user send commands back to their smart IoT devices. Graphs, bars, and charts present information for users in an understandable way. We can also send commands from a mobile phone app to sensor units to change settings.

IPv6 Addresses

IPv6 addresses are the backbone of the entire IoT system. The Internet is only concerned with IP addresses and not if the address to which it refers is a lamp or a refrigerator. IPv6 has as many as 3.4×10^{38} IP addresses. This will ensure that new IoT devices connected to the Internet in the future will be able to get a unique IP address. Figure 10.1 present the IoT architecture.

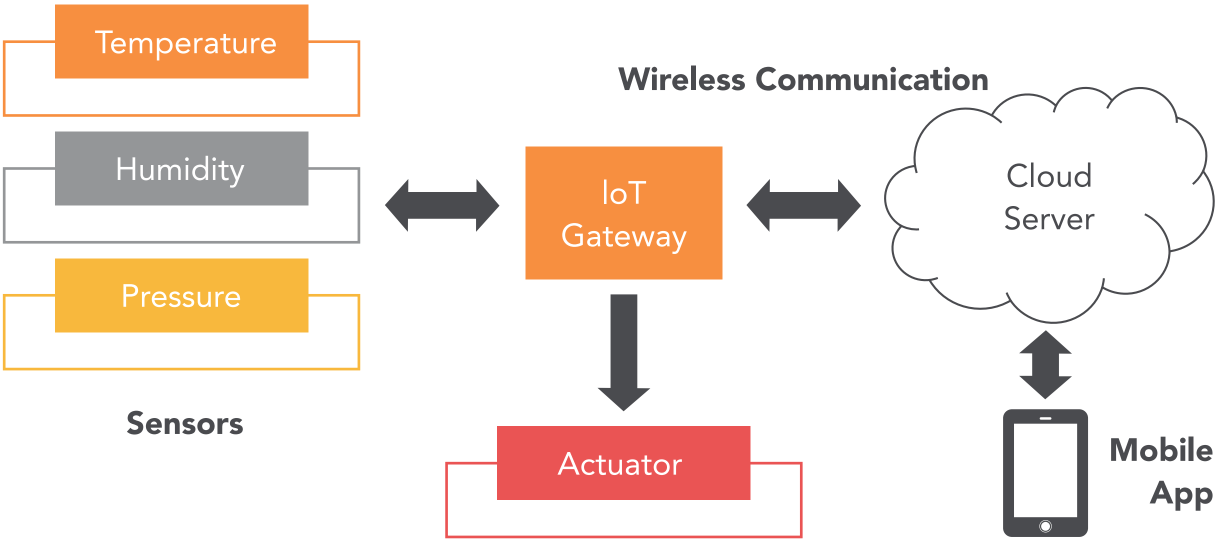


Figure 10.1 Internet of Things (IoT) architecture

10.2 SOME REQUIREMENTS FOR AN INTERNET OF THINGS ARCHITECTURE

There are some requirements that are specific to IoT devices and the environments that support them. Some requirements come from equipment and software with specific limitations. Other requirements come from the manufacturing and use of IoT devices. The requirements are more about traditional consumer product design than existing Internet approaches. In addition, there are several existing best practices for the server side and Internet connection that one must use.

The following list includes the general requirements for some important parts of the IoT:

- Connection and communication;
- Device management;
- Data collection, analysis and activation;
- Scalability; and
- Security.

Connection and Communication

Earlier protocols like HTTP are important for many IoT devices. However, HTTP and some other traditional Internet protocols can be a problem for two reasons. First, the memory size of the application may be a problem for small devices. The biggest problem is the requirement for low power consumption. To meet these requirements, IoT devices need a simple and small protocol. In addition, there are IoT devices that connect directly and those that connect via gateways. The devices that connect to the cloud via a gateway often require two protocols: one for connecting the device to the gateway and another to connect the gateway to the cloud.

Device Management

Many IoT devices are not properly managed, and this is often unfortunate. Active control of PCs, mobile phones and other devices is becoming increasingly important, and the same path is both likely and desirable for IoT devices.

The following list provides some desirable requirements for managing IoT devices:

- The ability to update the software on a device;
- Enable or disable certain hardware settings;
- External remote configuration of Wi-Fi, General Packet Radio Services (GPRS) or network parameters;
- The ability to disconnect a device that has been stolen;
- Remove and secure data from a stolen device;
- Track down a device that has been lost; and
- Update security information.

The above list is not exhaustive: it could be longer and include features that might not be required or possible for some devices.

Data Collection, Analysis and Activation

We want an IoT system to collect data from many devices. We want to save, to analyse and to process the data.

An IoT architecture is designed to handle many devices. If these devices produce constant data streams, then large amounts of data are generated. This phenomenon requires a highly scalable storage system that can handle different amounts of data and large data volumes. In some cases, reactions to incidents must be possible in almost real time, and this is a requirement for real-time analysis. In addition, the device must be able to analyse and to process data. In some cases, this endeavour will only be simple computation, but for devices that are more complex, we need to utilise programs that are more powerful to deal with incidents.

Scalability

A server architecture should preferably be highly scalable and be able to support millions of devices that constantly transmit, receive, and process data. However, an architecture with high scalability often has a high price in terms of hardware, software, and complexity. An important requirement of this architecture is to support the scaling of the use of a small number of devices to many devices. Elastic scalability and the ability to distribute data in a cloud infrastructure are crucial. The ability to use small, inexpensive servers is a requirement for making a low-cost and large-format architecture.

Security

Security is important in connection with the IoT. Indeed, IoT devices often collect personal data, and it is their job to bring real-world data to the Internet. This ability leads to three categories of risk:

- Risk associated with any Internet system, but that which IoT designers may not be aware,
- Specific risks unique to IoT devices and
- Security to protect against damage caused by, for example, abuse of actuators.

The first category contains simple things like accessing IoT devices. For example, a refrigerator that was connected to the Internet and had an unsecured Simple Mail Transfer Protocol (SMTP) server could have been used to send spam. The second category contains issues specifically related to IoT hardware – for example, many units do not have enough security. Many IoT devices do not have resources to support proper asymmetric encryption. Another example is the possibility that someone might attack the hardware to understand its security. These types of reverse engineering attacks are a problem compared with pure web solutions, where there is often no available code to attack.

Two very important specific issues for IoT security are concern about identity and access management. Identity is a problem where practice is often poor.

10.3 INTERNET OF THINGS PLATFORMS

There is a need for an infrastructure to handle data streams from millions of devices attached to the IoT. The architecture of this type of real-time data stream processing needs to be able to handle data import, processing, storage, and analysis of hundreds of millions of events per hour.

An IoT platform (IoT Framework) has three building blocks:

1. The thing,
2. The gateway and
3. Network and cloud computing IoT architecture.

An IoT platform is a multi-layer technology that enables straightforward provisioning, management, and automation of connected devices within the IoT system. It connects hardware, however diverse, to the cloud by using flexible connectivity options, enterprise-grade security mechanisms and broad data processing powers.

IoT platforms originated in the form of IoT middleware, the purpose of which was to function as a mediator between the hardware and application layers. Its primary tasks included data collection from the devices over different protocols and network topologies, remote device configuration and control, device management and over-the-air firmware updates.

Many different structures for IoT platforms have been proposed to improve the connection of devices to the cloud. The key to successful IoT projects is collaboration between different units, solid IoT frameworks and platforms that enable a powerful analysis. In addition, this platform must offer value-creating services that make it attractive for companies to use the platform.

10.4 A FOUR-STAGE ARCHITECTURE OF AN INTERNET OF THINGS SYSTEM

A common IoT infrastructure is a four-stage architecture. These four stages can be considered shares of a process. All four are built-in, mutually reinforcing and carry data from things via various networks to traditional data centres for processing. This phenomenon provides users with information. Figure 10.2 summarises the four stages.

Stage 1	Consists of sensors, actuators, and network services, often wireless.
Stage 2	Includes aggregation systems for sensor data and analogue-to-digital data conversion. Aggregation means that data are processed and merged so that the amount of data becomes smaller.
Stage 3	Edge-information technology systems perform a pre-processing of the data before moving it to a data centre in the cloud.
Stage 4	The data are analysed, managed, and stored on data centre systems in the cloud.

Figure 10.2 The four stages in an Internet of Things architecture

Stage 1. Sensors and Actuators

Sensors gather nearby data that could provide us with useful information. In a regular IoT system, a sensor can collect information and send it to a data centre for processing. In response to the data from the sensor, an actuator receives a command. Based on the data from sensors, actuators can act on the local environment.

The sensors provide the IoT data, and it is the information from the sensor data that makes the IoT system intelligent. Because data are central to the IoT, it is important to ensure that the data are accurate. Sensors and actuators may also have the task of ensuring the accuracy of the system.

Data processing can occur in each of the four stages of an IoT architecture. However, even if you can process data near the sensor, there will be limited processing power available near an IoT device. The central part of an IoT system is data, and you must choose whether data processing should be local at the sensor or in the cloud.

To get good information from the data, a comprehensive treatment is often required, and to get this you need to move the data to a data centre in the cloud. However, not all decisions can wait until they are processed in the cloud. You need to deal with some decisions in real time. For example, did a robotic arm cut an artery when performing an operation in a hospital? Will a car crash? You do not have time to send such data to the cloud. You need to process such data close to the sensor on the edge of the network for the fastest possible feedback.

Stage 2. The Internet Gateway

The data from the sensors starts in an analogue form. Further processing aggregates these data and converts them into digital data. Data acquisition systems (DAS) perform data aggregation and conversion. The Internet gateway receives the aggregated and digitised data and routes it via Wi-Fi, wired LAN or Internet to systems in stage 3 for further processing.

Systems in stage 2 are often located near sensors and actuators. For example, a pump may contain half a dozen sensors and actuators that feed data into a data aggregation unit, which also digitises the data. The IoT unit is usually physically connected to the pump. An adjacent gateway device or server will then process the data and pass it on to the systems in stage 3 or 4.

There will soon be large amounts of data from the analogue data streams coming from sensors. Therefore, the data must be processed in advance. Values from the physical world that one might be interested in measuring are temperature, movement, tension, and vibration, among others. Measuring these factors can generate large amounts of constantly changing data. For example, an aircraft engine can generate huge amounts of data in a 24-h period. Theoretically, there is no limit to the number of sensors that could feed data into an IoT system. In addition, an IoT system is always active. IoT data streams can be enormous: as much as 40 TB/s has been observed. Given this huge volume, it is best to pre-process the data. Another reason not to send all the data to the data centre is that analogue data has specific timing and structural features that require specialised software to process. It is best first to convert the data to the digital form, and that is what happens in stage 2.

Intelligent gateways can build on basic gateway functionality by adding such features as analysis, malware protection and data management services. These systems allow analysis of data streams in real time. Gateways are units at the end of the system (edge). DAS and gateway devices are used in a variety of environments, from the factory floor to mobile field stations, so these systems are usually designed to be portable, easy to deploy and robust enough to withstand variations in temperature, humidity, dust, and vibration.

Stage 3. Edge Computing

Edge computing is data processing at the end of a network, that is, near the sensors or the data source. The purpose of performing aggregation and analysis at or near the data source is to reduce data traffic between sensors and central data centres. Edge computing covers a wide range of technologies, including wireless sensor networks, mobile data collection, mobile signature analysis, networking, and processing.

While the IoT data are digitised and aggregated and then ready to send to a data centre, further processing could be required before the data are sent. This additional processing involves edge IT systems that perform further analysis. Edge IT processing systems can be located at external offices or elsewhere on the edge, but these are located at the facility or somewhere closer to the sensors – for example, in a switch cabinet.

Because IoT data can easily eat up network bandwidth and use up resources in data centres, it is best to have systems at the edge that can perform analyses that somehow reduce the burden on the core IT infrastructure. If you send all the data directly to the data centre, it will quickly require an enormous storage capacity. You will also encounter security issues, storage issues and delays. With an edge computing approach, you can pre-process the data, get meaningful results and then forward the data to the data centre. For example, instead of transmitting raw vibration data for pumps, you can aggregate and convert the data, analyse it and resend only projections when a device fails or needs service.

Stage 4. The Data Centre and the Cloud

Data that need a more thorough treatment, and where feedback does not need to be immediately, are forwarded to a physical data centre or cloud-based systems, where more powerful IT systems can analyse, manage, and store the data safely. It takes longer to get results when you wait for data at stage 4, but you can do a more thorough analysis and combine your sensor data with data from other sources for deeper insight. Stage 4 processing can take place locally, in the cloud or in a hybrid cloud system, but the type of processing performed at this stage remains the same regardless of the platform.

10.5 IOT GATEWAYS

An IoT gateway is a gateway to the Internet for an IoT unit. An IoT gateway is a physical device or computer program that connects embedded devices with sensors to the cloud. All data moving to the cloud or the opposite way go through a gateway. An IoT gateway is also referred to as an intelligent gateway.

Some sensors generate tens of thousands of pieces of data per second. The edge provides a place to pre-process the data locally near the sensor before the data are passed on to the IoT gateway and transported to the cloud. The aggregation, summarisation, and analysis of data at the edge minimises the volume of data sent to the cloud, and this approach can reduce response times and network transfer costs.

Another advantage of an IoT gateway is that it can provide additional security for the IoT network and the data it transports. Because the gateway manages information moving in both directions, it can secure data sent to the cloud and protect IoT devices from hacker attacks.

Gateways are an important part of the IoT communication structure. The task is to connect sensor units to remote data centres in the cloud. Therefore, a gateway must have protocols that can communicate data over the Internet.

An IoT gateway creates a bridge between IoT devices, sensors, equipment, systems, and the cloud. IoT gateway devices offer local processing and storage solutions, as well as the ability to control sensor units automatically, based on the data input from the sensors.

An IoT gateway performs multiple critical functions and has protocols for encryption, processing, managing, and filtering data. Figure 10.3 lists some of the benefits of using IoT gateways.

Benefit	
Reduce Network Traffic	An IoT gateway can process data at the edge of the network so that only intelligent data are sent to the cloud. Data processing at the edge means less traffic on the network.
Lower Cost	Endpoint IoT units do not need high processing power, memory, or storage space because the gateway does all of this for them.
Faster Systems	Faster and more advanced computing can reduce time significantly.
Minimise Risk	Gateways can isolate non-functioning devices and sensors before they cause major problems.

Figure 10.3 Some benefits of using Internet of Things (IoT) gateways

10.5.1 THE TASKS OF AN INTERNET OF THINGS GATEWAY

We will see the main reasons for introducing a gateway into an IoT architecture by discussing some of the key aspects of how the gateway architecture works.

First, sensors usually have very limited network connectivity capabilities. There is a need for a gateway that can provide sensors with a single point of contact with external networks using Wi-Fi, Global System for Mobiles (GSM) or another type of connection.

A gateway is not just an unintelligent proxy that transfers all data from sensors to back-end services. Sending all the information collected by sensors to a data centre will be very ineffective in terms of performance and network utilisation. An IoT gateway is required to pre-process information near the sensor unit before data are sent to the data centre. Such pre-treatment includes filtering and aggregation.

The gateway should also act as a place for local monitoring of a sensor unit area. You do not need to connect monitoring software to all sensors; it is easier to monitor only the gateway, which in turn is responsible for collecting all the necessary calculations from the sensors.

IoT gateways also perform several improvements on IoT systems.

A Gateway Can Provide a Security Layer

As the number of embedded devices and sensors grows, communication across public and private networks is also increasing. Communication between things, the gateway and the cloud must therefore be secure so that unauthorised individuals do not manipulate or hack data. This security usually employs a public key infrastructure (PKI) where everything that communicates gets an identity – a pair of cryptographic keys or a digital certificate that allows encryption of data. This process can be difficult to handle without using an IoT gateway.

A Gateway Can Perform Device Updates

Imagine you discover a security issue in one of your devices, or you notice that one of the sensors is too hot. Without a gateway, you need to make manual corrections because the devices and sensors have too little computing power to perform such tasks themselves. If the system has a gateway, data from the sensors is sent to the gateway and the gateway is configured to send firmware updates to all devices when necessary.

10.5.2 THE ARCHITECTURE OF INTERNET OF THINGS GATEWAYS

An IoT gateway is an important component of an IoT system. It should be efficient, secure, and easy to maintain. The typical architecture of IoT solutions is usually complicated. One of the most important factors that increases the complexity of the IoT systems is the back-end services in the data centre. IoT systems must handle a multitude of devices that are dispersed globally. Because the properties of these devices are very different from web clients, desktop clients and mobile clients, there is a need for an intermediate component that will act as a proxy between the sensor unit and the data centre. What we need is an IoT gateway.

Overview of the Gateway Architecture

The following gateway architectural chart (Figure 10.4) presents the most common architecture of the gateway when not equipped with sensors. The gateway software installed on the device is responsible for collecting data from the sensor, pre-processing the data, and sending the results to the data centre.

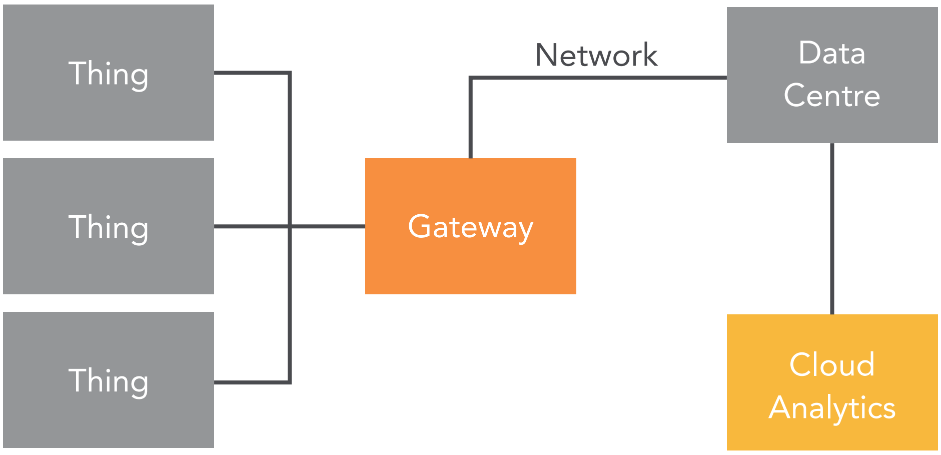


Figure 10.4 The connection between the things, the gateway, and the data centre

Summary

The gateway is a key component of all IoT solutions. When choosing the right hardware for an IoT solution, it is crucial to keep in mind that obtaining the right gateway software and management infrastructure is a factor that will have a major impact on overall maintenance cost for the system.

10.6 EDGE COMPUTING

Edge computing is a way to reduce data flow from IoT devices to the cloud. Before transmission from sensors to the cloud, the data are first pre-processed locally at the edge of the network, that is, near the sensor units. It can be beneficial in many situations to process data from IoT devices before sending it long ways to data centres in the cloud. By performing computing before sending the data over the Internet to a data centre, important data can be analysed in real time. Many organisations have a need for this approach: industry, health care, finance, and telecommunications.

How Edge Computing Works

Edge computing involves processing data locally near the sensor units. Then, the pre-processed data are sent for more processing and storage in a cloud data centre (Figure 10.5). One advantage of processing data locally via edge computing is that it reduces the amount of data sent to the cloud, reducing traffic to data centres in the cloud.

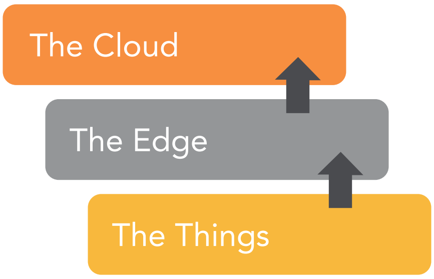


Figure 10.5 Edge computing allows data from the Internet of Things to be analysed at the edge before being sent to a data centre or the cloud

The IoT units transmit the data to a local unit, which performs data processing, storage, and networking. Data are first processed locally at the edge, and then all or a part of the data are sent to the corporate data centre or to a data centre in the cloud. Figure 10.6 provides definitions for commonly used edge computed expression.

Edge Computing Expressions	
Edge Devices	An edge device produces or collects data: sensors, industrial machines, or other devices.
Edge	The edge may be different depending on the context. In a company's information technology system, the edge can be a computer. In the telecommunications field, the edge can be a mobile phone or a mobile tower. In an Internet of Things (IoT) car system, the edge of the network can be a car. In the industry, there may be a machine on a factory floor.
Edge Gateway	An edge gateway is a transition between a device that performs edge computing and a larger network. Typically, an edge gateway will be a gateway to the Internet and the cloud.
Fat Client	A fat client is software that can perform some computing. This contrasts with a thin client, which only can transfer data.
Edge Computing Equipment	Various edge computing equipment such as devices, sensors and machines will easily work in a computer system. They must be connected to the Internet.
Mobile Edge Computing	This refers to the development of edge computing systems in telecommunication systems, especially 5G scenarios.

Figure 10.6 The definition of some commonly used edge computing expressions

Why Use Edge Computing?

Edge computing is ideal in a variety of circumstances. An example is when the connection to the cloud is poor and it is not possible for IoT devices to connect constantly to a central cloud. Other uses of edge computing are related to real-time processing of information. Edge computing provides faster computing because data do not need to be sent over a network to a data centre or to the cloud for processing. This is ideal in situations where a waiting time in milliseconds can be unsustainable, which is the case in real-time applications.

The following is an example of using edge computing. An oil rig in the sea can have thousands of sensors that produce large amounts of data, most of which have little significance. Perhaps the data only confirm that the systems are working properly. Data need not necessarily be sent over a network immediately after production. By just sending important data over the network, edge computing reduces the data flow that goes through the network.

Security can also play a role in wanting to use edge computing. Some argue that security is theoretically better locally because data are not transmitted over a network and stay closer to where they were produced. Indeed, the less data in a corporate data centre or a cloud environment, the less vulnerable the data.

The vast amount of data from the world's mobile users is changing the planning of network infrastructure. Hence, some believe that edge computing itself is less secure because the edge devices could be more vulnerable. Therefore, when designing edge computing, security is crucial. Data encryption and access control are important elements for protecting edge-computing systems.

10.6.1 5G AND EDGE COMPUTING

5G is the latest development in wireless and mobile communications; it stands for fifth generation networks. This technology has lower latency, higher bandwidth and improved privacy and security features than its predecessor (4G). In addition, it can connect far more devices. For the end user, this means better coverage and up to 10 times faster networking. This ability can be very valuable in cases where Internet speed is crucial, such as remote surgery, where doctors can use robots to operate on a human on another continent, or self-driving cars.

The combination of 5G networks and edge computing makes it possible both to develop new technological solutions that were previously too expensive or difficult to establish, and to improve existing technologies. Below are some examples of what these technologies can contribute:

- High-quality, problem-free video conferencing and streaming without delays;
- High degree of user mobility, such as fast Internet on planes and high-speed trains;
- Innovations in IoT that can contribute to fully automated production processes (lights-out manufacturing);
- Improved augmented reality (AR) technology;
- Ultra-reliable communication that can be used, for example, in health care services (remote surgery); and
- Control of robots in real time, autonomous cars, etc.

10.6.2 USE OF EDGE COMPUTING IN CARS

Edge computing is advantageous for autonomous cars. In autonomous cars – a data centre on wheels – edge computing plays a dominant role. Autonomous cars will generate large amounts of data and sending all of it to the cloud is uncertain, unnecessary, and impractical.

It is uncertain whether data should be sent to the cloud because events on the edge must be processed in real time with very low latency. A self-propelled car that sends all data to the cloud as it drives on streets and highways could lead to a disaster. An example is when a person enters the street in front of a car. In this situation, a short waiting time before a reaction is required: the car must brake immediately. There is no time to send the data to the cloud for processing. It is also unnecessary to send all the data to the cloud because most data have only short-term value. A person in front of a car is information that usually is uninteresting in retrospect. It is the reaction time that is important, not the data. Transporting large amounts of data generated from vehicles to the cloud is simply not practical.

Nevertheless, the cloud still has a role for data from autonomous vehicles. Data regarding how cars respond to an event can be valuable because it provides information on how the car system tackles dangerous situations and what is happening in the traffic.

10.6.3 EDGE COMPUTING OR CLOUD COMPUTING?

Connected devices collect and process data from the physical community to make life easier and better for us. Many companies use IoT data to understand their business better and to make good decisions. For example, a shipping company can place sensors on their pallets, crates, and containers to track geographic location, ambient temperature, pressure, and other environmental variables.

The IoT only grows and grows. With the rapid development of affordable sensors and IoT devices, many expect that a company's use of IoT will explode. This increased use leads to an important question: What is the best architecture to handle all this data? There are three options:

1. Local architecture,
2. Cloud-based architecture and
3. Hybrid architecture.

Local Internet of Things Architecture

A local IoT architecture uses edge computing where data are processed near the data source at the end of the network. The edge computing architecture provides information that can help companies provide real-time data responses. For example, on an oil rig, sensors can detect whether a defective valve poses a fire hazard. In such a case, delays could have adverse consequences. If data are to be sent long ways to a data centre and back before giving a warning to shut off the valve, it may be too late. By processing the data locally, time delays are reduced, and quick decisions could be made. Furthermore, a local architecture does not rely on Internet connections to data centres in the cloud. Moreover, companies seeking good data security favour local architectures. There are many uses where local architecture and edge computing are beneficial.

Cloud Architecture

A cloud based IoT architecture can be beneficial for organisations that have many connected devices where the desired information is based on a combination of internal and external data. For example, applications can benefit from understanding some data in relation to the aggregated view of all the data. Then, only one set of data will lose its meaning outside a larger context.

In addition, a cloud architecture offers more opportunities to collaborate and to interact with other IoT devices and cloud systems. This model provides far more architectural flexibility and influence of external data sources. The IoT distributions that utilise cloud architectures can be more effective because of the innovative and competitive offerings that only cloud providers can make available. Essentially, a cloud architecture could better secure IoT investments for organisations.

A Hybrid Architecture

The best approach is often one that effectively combines the processing of large core datasets locally at the edge and then sends a reduced set of aggregated data for processing in a remote data centre. As an example, smart cities that use car-parking sensors can process all sensor data near the garages and just send a summary of data regarding the number of vacancies in parking spaces to the cloud. This approach is not as expensive as sending all the data yet still provides useful information: drivers who are going to park need only know that there are vacancies in a garage. They do not necessarily have to know which exact parking spaces are available. In such cases, a hybrid architecture is ideal.

11 INTERNET OF THINGS COMMUNICATION

The Internet consists of many different technologies that communicate with equipment, anything from a single device to large platforms of embedded technologies and cloud systems connected in real time. IoT devices must be able to communicate with each other. Data must be collected and sent to a server infrastructure. This server infrastructure must be able to store and to share data, and possibly send it back to IoT devices or to users.

11.1 WIRED AND WIRELESS INTERNET OF THINGS SOLUTIONS

Wired and wireless have advantages and disadvantages when it comes to network connectivity. Understanding these benefits and drawbacks will help you make better decisions when implementing an IoT solution. About half of IoT devices are connected to wireless networks, and the other half are still wired.

Wired Solutions

IoT technology is deployed in many ways so that no single network solution is right. Solutions depend on the circumstances and where the devices are located. Some of the factors affecting the selection of the type of network are network range, network bandwidth, power usage, interoperability, intermittent connectivity, and security.

Wired networks represent an established technology, and it is easy to plug into them if you already have phone, power, and coaxial cable lines. Even wireless networks are usually connected to a wired network at some point; hence, the most used network is a hybrid of both wired and wireless network connectivity. Figures 11.1 and 11.2 present some benefits and disadvantages, respectively, of wired connections.

Benefits	
Reliability	Ethernet connections have been in existence much longer than Wi-Fi technology, which makes it much more reliable. They are less prone to dropped connections and are more reliable without constant debugging.
Speed	Wired connections are less affected by local factors like walls, floors, cabinets, the length of the room, interference from other electronic devices, etc. Hence, wired connectivity is faster than wireless. Wired data transmissions are not sensitive to distances and placement of devices does not have any adverse effect on the performance of the connection.
Security	Wired connections are usually housed behind your local area network (LAN) firewall; thus, you can have complete control of the communications system. This means there is no broadcasting data that can be hacked into.

Figure 11.1 Some benefits of wired connections

Disadvantages	
Cost	Wired connections are more expensive than wireless due to the cost of the wire and labour for installation. In the event of a damaged cable, the repair or replacement costs are also extremely high compared with the relatively low-maintenance nature of wireless networks.
Mobility	Wired networks would need to be buried in walls, floors, and ceilings to reach the sensors that need to be connected to it. Because the sensors are small and can be placed anywhere in a facility, it would sometimes be physically impossible to reach them.
Scalability	Building and extending a wired network requires planning and a budget to construct it. For wired systems, hardware needs to be purchased, installed, and configured before it can be fully operational. Scalability would be an issue not only for networks to be up and running quickly but also for planning and cost purposes.

Figure 11.2 Disadvantages of wired connections

Wireless Internet of Things Implementation

As most wired networks tend to be large and expensive, wireless IoT implementations are the common solution. Setting up and configuring a wireless network is a simple process: it can be up and running in very short time. With the evolution of network technologies, we see a wide range of solutions like RFID, Bluetooth, Wi-Fi, ZigBee, Z-Wave or ultra-wideband (UWB).

There are four common communication models used by IoT:

1. Device to device,
2. Device to cloud,
3. Device to gateway and
4. Back-end data sharing.

The type of wireless network implemented will depend on the communication model.

Device to device uses Bluetooth, Z-Wave or Zigbee as it involves transmitting small amounts of data.

Device to cloud uses Wi-Fi or cellular technology. Cloud connections allow users to obtain access to the device remotely.

Device to gateway uses the network of your smart devices like a smartphone or a smartwatch. Examples of this are fitness trackers that upload data into your mobile app.

Back-end data sharing essentially extends the single device-to-cloud communication model so that IoT devices and sensor data can be accessed by authorised third parties. Under this model, users can export and analyse smart object data from a cloud service in combination with data from other sources and send it to other services for aggregation and analysis. This can use any network connectivity like Wi-Fi, cellular or even satellite. It all comes down to the use case of your business. Figures 11.3 and 11.4 list some advantages and disadvantages, respectively, of wireless connections.

Advantages	
Scalable	Wireless networks do not require any hardware installations. They typically involve configurations and can be up and running in a short time. They can also be extended very easily without considerations of obstructions in the facility. Newer wireless technologies use plug and play including auto-discovery that reduces installation times.
Cost-Effectiveness	Due to advancement in wireless technology as well as the number of manufacturers, the cost of wireless networks has decreased over the last few years. Moreover, most wireless sensors come with nodes that can be extended by adding additional nodes as required.

Figure 11.3 Advantages of wireless connections

Disadvantages	
Interference	Electronic devices in the vicinity of wireless networks can easily interfere and can cause loss of the connection or reduced quality of the connection. This can lead to loss of productivity until the issue is identified and fixed.
Slower speed	When dealing with real-time data, it is imperative that data are transmitted and available as fast as possible. Wireless networks are susceptible to increased latency and signal interference that impacts the speed and consistency of the data.

Figure 11.4 Disadvantages of wireless connections

Conclusion

It is important to know the advantages and disadvantages of wired compared with wireless solutions: the right solution often depends on your business use case. Understanding the specific needs of your facility and how continuous monitoring will help reach your goals is the most important aspect when weighing the advantages and disadvantages of wired versus wireless network solutions.

11.2 SOME WIRELESS CONNECTIVITY TECHNOLOGIES

IoT devices can communicate with each other in many ways. Home and offices often use standard Wi-Fi or Bluetooth Low Energy. Ethernet is also used if the devices are not particularly mobile. Other devices will use 4G, 5G or satellite connections to communicate. 5G is the successor of 4G and is a standard for high-speed telecommunications. The large number of different communication options for the IoT has led to a need for better standards. Communication standards should be recognised and should work as well as Wi-Fi has.

As the IoT evolves, it is likely that less data will be sent to the cloud for processing. To keep costs down, edge computing is more likely to be used. Data processing is then done locally near the unit and only aggregated data are sent to the cloud.

The old standards for Wi-Fi communication named 802.11 will continue to play a role. It has good speed and low cost, making it attractive for many applications. It can be used over larger areas. It requires one or more available hotspots, which are easy to set up.

There are several brands of wireless technologies (Figure 11.5). ZigBee and other short-range wireless devices can be used in wireless sensor networks. Z-Wave can also be used in these networks. In addition, Bluetooth can create a personalised network (PAN) and thus adapt to specific applications. For example, Bluetooth Low Energy 4.0 and similar ultra-low energy versions fit particularly well for applications in medicine and physical exercise.

Figure 11.5 Some brands of wireless technologies

ZigBee and other 802.15.4 wireless devices are very good options thanks to their low power consumption and ability to form wireless sensor networks. The 6LoWPAN Internet Engineering Task Force (IETF) standard allows you to connect billions of devices to the Internet by transmitting IPv6 packets with the 802.15.4 low power standard.

Two wireless technologies that has become a success in the IoT are RFID and near-field communication (NFC). NFC as well as RFID use the wireless area for communication. The area is limited to centimetres, but it is still useful for many applications. NFC-connected smartphones can replace credit cards to make payments: just tap the vendor's NFC reader with your mobile phone to approve a purchase. Some smartphones already integrate NFC, but this is still in the early stages.

The IoT is a network of things with the possibility to communicate with each other. The most important issue here is connection among devices. We can connect the wireless communication protocols to the following six standards:

- Satellite,
- Wi-Fi,
- Radio frequency (RF),
- RFID,
- Bluetooth and
- NFC.

Below, we will provide a brief overview of each of these communication techniques.

Satellite

Satellites communicate by using radio waves to send signals to the antennae on Earth. The antennae then capture and process the information coming from those signals. Satellites make communication from a mobile phone to an antenna possible at a distance of approximately 15–25 km. This communication is called GSM, GPRS, Code Division Multiple Access (CDMA), 2G (GSM), 3G, 4G (LTE), 5G, EDGE and more, based on the connection speed. Examples of satellite connectivity are measuring equipment that sends data to a remote server, or cars connected to the Internet.

An advantage of satellite communication is stable connectivity and universal compatibility. One disadvantage of satellite communication is that there is no direct communication from a smartphone to a device: the communication must go via satellite. Satellite communications also have had high monthly costs, high power consumption and high response time.

Satellite is useful for communications that transmit small amounts of data and has been primarily suited for industrial purposes. Soon, where satellite communication prices should fall gradually, the use of satellite technology could be much more viable and appropriate for consumers.

Radio Frequency

An RF signal refers to a wireless electromagnetic signal that is used as a communication form when discussing wireless electronics. Radio waves are a form of electromagnetic radiation, with RF ranging from 3 kHz to 300 GHz.

RF communication is probably the simplest form of communication among devices. Protocols such as ZigBee or Z-Wave use a low-frequency RF radio that is built-in or subsequently mounted in electronic devices and systems. Z-Wave has a range of approximately 30 m. The used RF band is specific to different countries. ZigBee is based on the IEEE 802.15.4 standard, but its low power consumption limits transmission distances to a range of 10–100 m.

An example of an RF connection would be the distance you have to the TV because it uses an RF that allows you to switch channels with a remote control. Other examples include wireless light switches, electrical meters, traffic control systems and other consumer and industrial equipment that require short-range and low-speed wireless data transfer.

RFID is a method for storing and retrieving data using small devices called RFID tags. An RFID tag is an integrated circuit that can be attached to or built into a product, animal, or person. RFID chips contain antennae that enable them to receive and to respond to RF signals from an RFID transmitter. Passive chips respond with a weak radio signals and need no power source, while active chips transmit a more powerful response signal over a slightly greater distance and require a power source. The known applications for RFID are tracking, logistics, clothing, passports, ticketing, electronic payment, container terminals, product protection/burglar alarm, evacuation systems, access control systems and animal identification.

RFID is a wireless use of electromagnetic fields to identify objects. Usually, you would install an active reader or read codes containing stored information, mostly authentication responses. The short range for RFID is approximately 10 cm while the long range can extend up to 200 m.

The advantages of RFID are that it does not require power and that it is an established and widely used technology. The disadvantages of RFID are that it is very uncertain, it has a running cost per card and it is not compatible with mobile phones.

Examples of the use of RFID are data collection in a factory, animal identification, access duty and access to buildings. RFID tags are also attached to a product so that the production process can be traced through the assembly line. As an illustration, medicines can be traced through department stores.

Near-Field Communication

NFC is a technology that enables easy and secure two-way transfers between electronic devices, especially mobile phones, enabling consumers to make wireless payment transactions, access digital content and connect to electronic devices. With this technology, the capacity of wireless card technology has been expanded and devices can share information when they are less than 4 cm apart.

NFC uses electromagnetic induction between two antennae located within each other's adjacent fields, effectively forming an air core shaped transformer. NFC handles stickers, car keys or battery-powered cards. NFC peer-to-peer communication is possible if both devices are powered.

NFC offers a low-speed connection with a single setup. It has a short range and supports encryption, which may be more suitable than before. A disadvantage of NFC is its short range, and it may not be possible to use in many situations.

NFC devices can act as electronic identity documents and key cards. NFC devices can be used in wireless payment systems, such as those used in credit cards and electronic cards and allow mobile payment by replacing or completing these systems.

11.3 FOUR TYPES OF INTERNET OF THINGS WIRELESS NETWORKS

In the following section, we describe the four general types of IoT networks and some IoT wireless protocols within each category.

11.3.1 CELLULAR

Cellular networks use the same mobile networks as smartphones to allow IoT devices to communicate. Because these networks were originally designed for power-hungry devices like smartphones, they were not always considered the best fit for IoT devices. The cellular industry has developed new technologies that are more appropriate for IoT use cases. Today, this type of wireless network is very popular and is considered a reliable and secure method of IoT connectivity. Cell service is available in most locations, and this type of network covers a very large area.

However, cell connectivity is often not available in places that most need monitoring sensors – for example, inside utility closets, basements, elevator shafts, etc. Another IoT wireless technology class, LPWAN, might be better for these locations. Even though cellular connectivity is now less expensive and more power efficient than traditional telecom standards, cellular-connected IoT devices still require a great deal more power and energy than some other types of wireless networks.

Two cellular IoT wireless protocols that have dominated are LTE-M and NB-IoT. However, the IoT is rapidly developing and expanding. 5G will increase cellular bandwidth by huge amounts, making it much easier for the IoT to network large numbers of devices together. 5G will quickly become the new standard for cellular networks. According to reports, 5G will be 10 times faster than current LTE networks. This increase in speed will allow IoT devices to communicate and share data faster than ever. In addition to the increase in speed, 5G networks will operate more reliably, creating more stable connections. Having a reliable and stable network condition is extremely important for any IoT, but especially for connected devices like locks, security cameras and other monitoring systems that depend on real-time updates. With the ability of a 5G network to handle more connected devices, consumers will benefit from greater reliability of their connected devices.

11.3.2 LOCAL AND PERSONAL AREA NETWORKS

Networks that cover short distances are called PAN and LAN; they are cost-effective, but the transfer of data can sometimes be unreliable.

Wireless PAN and LAN technologies that are commonly incorporated into IoT connectivity solutions are Wi-Fi and Bluetooth. Wi-Fi can be used for applications that run in a local environment, or in a distributed setting if there are multiple access points integrated into a larger network. One downside of Wi-Fi is that it works only if the signal is strong, and you are close to the access point. Moreover, Wi-Fi is generally power-hungry, but it is possible to operate it in a way that is a little more power efficient. Bluetooth Low Energy is a more energy-efficient wireless network protocol. If you are not receiving data constantly, a single battery running Bluetooth Low Energy could last up to 5 years. However, compared with Wi-Fi it is slower to transmit and is more limited in the amount of data it can send. Both Wi-Fi and Bluetooth are easy to connect in most cases, although Wi-Fi does have some security challenges that may be difficult to overcome.

11.3.3 LOW-POWER WIDE-AREA NETWORKS

IoT devices that run on LPWAN send small packets of information infrequently and over long distances. This type of wireless network was developed in response to the early challenges of cellular connectivity. Supporters of LPWAN position it as having a longer range than Wi-Fi and Bluetooth but using less power than cellular. Sigfox built the first LPWAN network in France and is considered the driving force behind its growth.

A well-known and commonly used IoT network protocol in this category is the long-range wireless area network (LoRaWAN), which runs on the long-range (LoRa) communication network. The advantages of LoRaWAN for IoT devices are its low power requirement and relatively low-cost chipsets. In addition, under the right conditions, a single base station or gateway running on a long-range network can provide service to a very large area; a few kilometres in dense urban areas and up to 15–30 km in rural areas.

11.3.4 MESH NETWORKS

Wireless mesh networks, an emerging technology, may bring the dream of a continuous connected world into reality. In mesh networks, all the sensor nodes cooperate to distribute data amongst each other to reach the gateway.

Zigbee is one example of an IoT wireless mesh network technology. Mesh networks are very short range and may require extra sensors throughout a building or the use of repeaters to get the coverage your application needs. Moreover, the nature of the way these networks communicate can result in high power consumption, especially if you need instant messaging, such as for a smart lighting application. However, mesh networks are also robust, able to find the fastest and most reliable paths to send data and easy to install, making them a popular choice for in-building use.

What is a Wireless Mesh Network?

A mesh network comprises interlocked routers called nodes, or points (Figure 11.6). These nodes work with each other to supply Internet coverage over a broad area. Each node spreads the radio signal a little further than the last, minimising the possibility of dead zones.

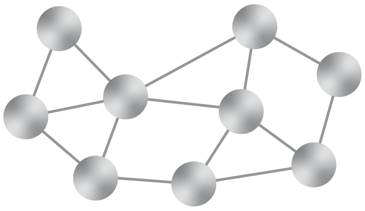


Figure 11.6 The structure of a mesh network

IoT devices often communicate with each other as in a mesh network, and not always to a server in the cloud. An example is streetlights that send signals to each other.

Why Use Mesh Networking for the Internet of Things?

While wireless mesh networking technologies has been around for some time, only recently has the power of mesh reached a point of maturity alongside high availability from chip and silicon vendors. With newer approachable costs, wireless mesh networking has become ideal for IoT builders. Moreover, with the rise in connected homes and industry support on open-source resources like Thread, mesh networking is now truly accessible while having a low enough cost that allows scaling up for production. As such, wireless mesh networking is becoming a much more viable, real choice for industrial and commercial IoT applications. It can provide additional services in a system in which extending a connection between two nodes is limited.

Smart Cities

In a wireless mesh network, the network connection is spread out among dozens or even hundreds of wireless mesh nodes that talk to each other to share the network connection across a large area. Wireless mesh networks can easily, effectively, and wirelessly connect entire cities using inexpensive, existing technology.

Wireless mesh networking is great for extending radio signals through parking garages, campus grounds, business parks and other outdoor facilities. Parking garages that utilise space availability checkers benefit greatly from mesh networks because they can extend the signal throughout the whole space and be able to communicate when other clients have taken a spot.

Health Care Equipment

Wireless mesh networks can help monitor and locate medical devices quickly. They can also act as a backup for medical equipment that always needs to remain online. If one node loses connectivity, another node can step in to keep the connection alive.

Industrial Internet

Wireless mesh networking can help you track pallets and monitor large physical objects with a highly reliable wireless network. With wireless mesh networks, you can easily track key data across your factory floor and across multiple locations to identify issues before they happen.

Advantages with Mesh Networking

Mesh networking provides Internet in areas that do not have ethernet connections or that are too far from the primary router. A single point of failure is no problem, which is the issue in star topologies and bus topologies. If one node can no longer operate, the mesh network can reroute, which enables it still to communicate between the remaining nodes. Taking a mesh network down is impossible unless there is worldwide catastrophe that wipes out all electronic devices in the world.

A mesh network works with minimal infrastructure and can therefore be deployed faster at a lower cost than traditional infrastructure. Because the devices in a mesh network can retransmit signals further, they can connect thousands of sensors over a wide area. A mesh network is also suitable for connecting devices in remote areas.

There is no centralised authority in a mesh network. Everything operating within the local network can run smoother because the nodes can communicate with each other instead of having to communicate with the central router.

Installation and management of most mesh networks is very easy because they are controlled with a companion mobile app. Extending the mesh network with more nodes is as easy as plugging the nodes into a power outlet and updating the app. Setup might cost less than a traditional network if you consider the ease of adding nodes and the fact that very little installation needs to take place; you do not have to run any networking cables.

Disadvantages with Mesh Networking

Mesh networks are tough to manage and to troubleshoot. For big networks, one needs a strong mesh technology to make it worthwhile, and this can be difficult to find.

Battery life affects the availability of the nodes. For example, a node with a dead battery could disrupt the network, causing more routing overhead and less reliability.

A mesh network system typically costs more than a traditional router. The cost of deployment can sometimes be problematic in certain scenarios. However, it can be redeemed by downloading a software development kits (SDK), which enables you to become a participant node in the entire mesh instead of building it from scratch.

Market and regulatory forces make mesh networking difficult to deploy.

Mesh networks can replace Wi-Fi providers, phone carriers and other intermediaries that provide connectivity to people. Consequently, intermediaries do not want to support this technology financially.

11.4 INTERNET OF THINGS PROTOCOLS AND STANDARDS

Connecting components to the Internet requires protocols that allow devices and servers to communicate with each other. Many connections are required to link a widely branched and split IoT network.

A protocol is a standard set of rules that allow electronic devices to communicate with each other. Communication between IoT devices must use protocols of the same standard. Devices that connect must be able to communicate with each other to transfer data. With different standards, the devices will have difficulty communicating with each other. Compatible standards are therefore important for creating a large and reliable IoT network. Today, there are several IoT communication protocols and standards designed to simplify IoT design. Today's Internet supports hundreds of protocols. The IoT will support hundreds more. It is important to understand for what each of these important protocols is designed.

IoT protocols and standards are broadly classified into two separate categories. These are:

1. IoT data protocols and
2. Network protocols for the IoT.

11.4.1 INTERNET OF THINGS DATA PROTOCOLS

IoT data protocols are used in the presentation/application layers. IoT data protocols are used to connect low-power IoT devices. They provide communication with hardware on the user side, without the need for any internet connection.

The connectivity in IoT data protocols and standards is through a wired or cellular network. Some examples of IoT data protocols are described below.

HTTP is an application-layer protocol for transmitting hypermedia documents, such as Hypertext Markup Language (HTML). It was designed for communication between web browsers and web servers, but it can also be used for other purposes.

A question is why are there any protocols outside of HTTP to transport data across the wide-area network (WAN)? HTTP has provided significant services and abilities for the Internet for over 30 years, yet it was designed and architected for general purpose computing in client/server models. IoT devices can be very constrained, remote and bandwidth limited. Therefore, more efficient, secure, and scalable protocols are necessary to manage a plethora of devices in various network topologies such as mesh networks.

The HTTP protocol is not preferred as an IoT standard because of its cost, battery life, huge power consumption and weight issues. Nevertheless, it is still used within some industries. For example, manufacturing and 3D printing rely on the HTTP protocol due to the large amounts of data it can publish. It enables PC connection to 3D printers in the network and printing of 3D objects.

A Constrained Application Protocol (CoAp) is an application layer protocol. It is designed to address the needs of HTTP-based IoT systems. HTTP is the foundation of data communication for the World Wide Web.

While the existing structure of the Internet is freely available and usable by any IoT device, it is often too heavy and power-consuming for most IoT applications. This has led to many within the IoT community dismissing HTTP as a protocol not suitable for IoT. However, CoAp has addressed this limitation by translating the HTTP model into usage in restrictive devices and network environments. It has incredibly low overheads, is easy to employ and can enable multicast support. Therefore, it is ideal for use in devices with resource limitations, such as IoT microcontrollers or wireless sensor nodes. It is traditionally used in applications involving smart energy and building automation.

Message Queue Telemetry Transport (MQTT) is a lightweight IoT data protocol aimed at collecting data. As the name conveys, the main purpose is telemetry or remote monitoring. The goal is to collect data from many devices and transport the data to an IT infrastructure. The purpose of MQTT is that large networks with many small devices can be monitored or controlled from the cloud. MQTT relies on TCP for data transmission. A variant, MQTT-SN, is used over other transports such as User Datagram Protocol (UDP) or Bluetooth.

The features of MQTT make it an excellent option for sending high volumes of sensor messages to analytics platforms and cloud solutions. MQTT makes it possible to monitor a large oil pipeline for leaks or vandalism. Thousands of sensors are connected to a single main unit for analysis. When the system finds a problem, it can do something to solve this problem. Examples regarding the use of MQTT include monitoring power consumption and lighting control.

The Advanced Message Queuing Protocol (AMQP) is an open standard for passing business messages between applications or organisations. It connects systems, feeds business processes with the information they need and reliably transmits onwards the instructions that achieve their goals.

AMQP is focused on not losing messages. It uses TCP for communication, which provides strictly reliable point-to-point connectivity. Furthermore, the end recipient must confirm receipt of each message. True to its origin in the banking sector, AMQP focuses middleware on tracking all messages and ensuring that each is delivered as desired, regardless of error or restart.

AMQP is mainly used in business communication. It usually defines devices as mobile phones that communicate with back-office data centres. In the IoT context, AMQP is most appropriate for server-based analysis functions.

XMPP

Extensible Messaging and Presence Protocol (XMPP) is an open technology for real-time communication that provides a wide range of applications including instant messaging, presence, voice and video calls, collaboration, and generalised routing of Extensible Markup Language (XML) data. XMPP was designed to connect people via text messages. The name highlights the purpose of the use: presence means people are closely involved.

XMPP uses the text format of XML that is suitable for person-to-person communication. Like MQTT, it uses TCP or maybe HTTP over TCP. One of the strengths of XMPP is an address system of the form named@domain.com, which helps connect the devices in the large Internet system. XMPP offers an easy way to add addresses to devices. XMPP is a good way to connect, for example, a thermostat at home to a web server so you can access it from your mobile phone. The power of addressing, security and scalability make XMPP ideal for consumer oriented IoT applications.

DDS The Proven Data Connectivity Standard for the IoT

Data Distribution Service (DDS) is an open standard for real-time applications. The Object Management Group (OMG) DDS is a middleware protocol and API standard that provides data connectivity, extreme reliability, and a scalable architecture to meet IIoT application requirements. Unlike MQTT and XMPP, DDS refers to devices that directly use data from other devices. It distributes data to other devices, and it is sometimes called middleware or a connectivity framework.

High-performance integrated device systems use DDS. It is the only technology that provides the flexibility, reliability and speed that are needed to build complex real-time applications. Applications include military systems, wind farms, hospital integration, medical imaging, real estate tracking systems and car testing and security.

WebSocket was initially developed back in 2011 as part of the HTML5 initiative. Via a single TCP connection, messages can be sent between the client and the server.

Like CoAp, WebSocket's standard connectivity protocol helps simplify many of the complexities and difficulties involved in the management of connections and bi-direction communication on the Internet. It can be applied to an IoT network where data are communicated continuously across multiple devices. Therefore, you will find it used most in places that act as clients or servers, including runtime environments or libraries.

11.4.2 NETWORK PROTOCOLS FOR THE INTERNET OF THINGS

Now that we have covered IoT data protocols, we will look at the different network protocols for the IoT. In this context, network protocols are used in datalink/physical layers. IoT network protocols are used to connect devices over a network. These sets of protocols are typically used over the Internet. Below are some examples of various IoT network protocols.

There is no denying that Wi-Fi is the most well-known IoT protocol on this list. However, it is still worth explaining how the most popular IoT protocol works.

To create a Wi-Fi network, you need a device that can send wireless signals. These include:

1. Telephones,
2. Computers and
3. Routers.

Wi-Fi provides an Internet connection to nearby devices within a specific range. Another way to use Wi-Fi is to create a Wi-Fi hotspot. Mobile phones or computers may share a wireless or wired Internet connection with other devices by broadcasting a signal.

Wi-Fi uses radio waves that broadcast information on specific frequencies, such as 2.4 or 5 GHz channels. Furthermore, both frequency ranges have several channels through which different wireless devices can work. This prevents the overflow of wireless networks.

A range of 100 m is typical of a Wi-Fi connection. The most common is limited to 10–35 m. The main impacts on the range and speed of a Wi-Fi connection are the environment and whether it provides internal or external coverage.

Bluetooth®

Bluetooth is an important communication technology over short distances and has been given a prominent role in computing and many consumer products. It is expected to be central to portable products that connect to the IoT, but also in many cases via smartphone. Bluetooth is used in many products such as phones, tablets, media players and robot systems. The technology is very useful for transferring information between two or more devices that are close to each other and in low bandwidth situations. Bluetooth is often used to transfer audio data to phones with Bluetooth hearing clocks, or file transfer for handheld computers.

Compared with other IoT network protocols listed here, Bluetooth tends to frequency hop and has a generally shorter range. However, it has gained a huge user base due to its integration into modern mobile devices such as smartphones and tablets, as well as wearable technology, such as wireless headphones.

Standard Bluetooth technology uses radio waves in the 2.4 GHz industrial, scientific, and medical (ISM) frequency band and is sent in the form of packets to 1–79 channels. However, the latest Bluetooth 4.0 standard has 40 channels and a bandwidth of 2 Mhz. This guarantees a maximum data transfer of up to 3 Mb/s. This new technology is otherwise known as Bluetooth Low Energy and can be the foundation for IoT applications that require significant flexibility, scalability, and low power consumption.

zigbee

ZigBee is a high-level communication protocol used to create PAN with small, low-power digital radios, such as for home automation, medical device data collection and other low-power, low-bandwidth needs, designed for small-scale projects that require wireless connection. ZigBee is widely used in home automation and in industrial environments. ZigBee PRO and ZigBee Remote Control, which are among the available ZigBee profiles, are based on the IEEE802.15.4 protocol, which is a wireless network technology for the industry. This protocol operates at low data rates over a limited area and within a 100-m range that fits in a home or building. Zigbee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach ones that are more distant.

ZigBee has some significant benefits in complex systems. It offers low power consumption, high security, robustness, and high scalability and is well positioned to utilise wireless control and sensor networks in M2M and IoT applications. The latest version of ZigBee is 3.0, which is essentially the unification of the various ZigBee wireless standards in a single standard.

ZWAVE

Z-Wave is a popular IoT protocol. It is a wireless, RF-based communication technology that is primarily used for IoT home applications. It is designed primarily for home automation and for products such as lighting control, security systems, thermostats, locks, and garage door openers, among many others.

It operates on the 800–900 MHz RF. Zigbee operate on 2.4 GHz, which is also a major frequency for Wi-Fi. By operating in its own range, Z-Wave rarely suffers from any significant interference problems. Of note, the frequency on which Z-Wave devices operate is location dependent, so make sure you buy the right one for your country. While Z-Wave is an impressive IoT protocol, like ZigBee, it is best used within the home and not within the business world.

LoRaWAN™

LoRa is a long-range wireless communication protocol. Because LoRa defines the lower physical layer, the upper networking layers were lacking. LoRaWAN was developed to define the upper layers of the network. LoRaWAN is like Sigfox and Neul and is designed to provide low-power WAN functions that are particularly needed to support low-cost mobile phones with secure two-way communication in the IoT.

LoRaWAN is a media access control (MAC) IoT protocol. LoRaWAN allows low-powered devices to communicate directly with Internet-connected applications over a long-range wireless connection. Moreover, it has the capability to be mapped to both the second and third layer of the OSI model. It is implemented on top of LoRa or Frequency-shift keying (FSK) modulation for ISM radio bands.

12 PLATFORMS FOR THE INTERNET OF THINGS

The IoT cannot function without software known as an IoT platform or an IoT cloud platform. As a form of middleware, an IoT platform sits between the layers of IoT devices, IoT gateways and applications.

IoT platforms have become a backbone of professional IoT deployments. IoT platforms enable the realisation of IoT projects and build IoT solutions faster, cheaper, and better. Their essential features and capabilities are on the level of connectivity and network management, device management, data acquisition, processing analysis and visualisation, application enablement, integration, and storage.

IoT platforms can be segmented into four distinct layers, each of which can comprise several components and/or subcomponents. The four main layers are:

1. Application management – providing the ability to develop rapidly, to test and to manage effortlessly IoT applications;
2. Data management – providing the ability to ingest, to store and to analyse data from IoT devices;
3. Telco management – providing telecommunications companies the ability to manage the connectivity to IoT devices at scale; and
4. Device management – providing the ability to configure, to monitor and to manage IoT devices remotely, including updates via networks.

Developing for the IoT is a complex task, and nobody wants to do it from scratch. IoT data platforms offer a jumping-off point by combining many of the tools needed to manage a deployment from device management to data prediction and insights into one service.

For developers, an IoT platform provides a set of ready-to-use features that greatly speed up development of applications for connected devices as well as take care of scalability and cross-device compatibility.

IoT platforms help:

- To connect hardware, such as sensors and devices;
- To handle different hardware and software communication protocols;
- To provide security and authentication for devices and users;
- To collect, to visualise and to analyse data that the sensors and devices gather; and
- To integrate all the above with other web services.

IoT platforms are the support software that connects everything in an IoT system. An IoT platform facilitates communication, data flow, device management and the functionality of applications.

12.1 SOME INTERNET OF THINGS PLATFORMS

There are now several hundred IoT platforms in the rapidly changing platform vendor market, and the IoT platform market is exploding. MarketsandMarkets forecasts the global IoT cloud platform market size to grow from USD 6.4 billion in 2020 to USD 11.5 billion by 2025, at a compound annual growth rate of 12.6% during the forecast period. The following section includes a description of some IoT platforms (listed alphabetically).

Amazon Web Services (AWS) IoT Core lets you connect IoT devices to the AWS cloud without the need to provision or to manage servers. AWS IoT Core can support billions of devices and trillions of messages and can process and route those messages to AWS endpoints and to other devices reliably and securely.

Arduino IoT Cloud is an application that helps makers build connected objects in a quick, easy, and secure way. You can connect multiple devices to each other and allow them to exchange real-time data. You can also monitor them from anywhere using a simple user interface. Arduino's open-source approach means you control how your system connects to the cloud. Arduino IoT Cloud is the simplest path for visual creation of cloud and device software, with webhooks to trigger events or store data with third party web services.

Google Cloud Platform

Google Cloud IoT is a complete set of tools to connect, to process, to store and to analyse data both at the edge and in the cloud. The platform consists of scalable, fully managed cloud services and an integrated software stack for edge/on-premises computing with machine learning capabilities for all your IoT needs.

IBM Watson

The IBM Watson IoT Platform is a fully managed, cloud-hosted service designed to make it simple to derive value from your IoT. It provides capabilities such as device registration, connectivity, control, rapid visualisation, and storage of IoT data.

What is KAA?

The Kaa IoT Platform is an enterprise-grade IoT platform built on a modern cloud-native architecture and a fully customisable feature set. Based on flexible microservices, Kaa easily adapts to almost any need and application. It can scale from a tiny start-up to a massive corporation and supports advanced deployment models for multi-cloud IoT solutions. However, you can also use it to put together a smart thermostat for your living room.

Microsoft Azure

Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centres. It provides software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS), and supports many different programming languages, tools, and frameworks, including both Microsoft-specific and third-party software and systems.

ORACLE
INTERNET OF THINGS

Oracle IoT Cloud Service is a managed PaaS cloud-based offering that helps you make critical business decisions and strategies by allowing you to connect your devices to the cloud, analyse data from those devices in real time and integrate your data with enterprise.

salesforce IoT cloud

IoT Cloud is a platform from Salesforce.com that is designed to store and process IoT data. In another context, IoT Cloud can provide business users with a much more comprehensive and integrated perspective on customers, without requiring technical expertise or the services of a data analyst.

ThingWorx
A PTC Business

The ThingWorx platform is a complete, end-to-end technology platform designed for the IIoT. It delivers tools and technologies that empower businesses to rapidly develop and deploy powerful applications and AR experiences.

12.2 WIRING THE INTERNET OF THINGS

Wiring together things in the IoT can be challenging and laborious. For that reason, tools have been developed to help in this endeavour. One of the challenges of the IoT is stepping into an object-oriented world and understanding how to link together so many disparate objects that speak different languages. The average IoT engineer is not interested in diving into the coding necessary to drive these interactions; they want to be able to pull in operational data quickly. This is where a tool that makes easily wiring together the IoT is incredibly valuable.

Node-RED

One tool that continues to make IoT and industrial solutions easier is Node-RED. Designed and built by IBM, Node-RED is a free, open-source logic engine that allows programmers of any level to interconnect physical I/O, cloud-based systems, databases, and APIs. Users interact with Node-RED via a browser-based flow editor that makes multi-device and multi-API integrations as easy as linking together nodes into easily understood flows.

Node-RED is a programming tool for wiring together hardware devices, APIs, and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

Node-RED reduces the need to write code, lowering the technical bar and allowing those interested in developing for the IoT to focus on the creating, rather than on the doing. It is surprising how many different applications for which people are using Node-RED, including schools teaching kids to code using Node-RED due its ease-of-use. While Node-RED is an incredibly useful tool for wiring together the IoT, it has applications far beyond the IoT. It can be used as a generic event-processing engine. For example, you can use it to listen to events from HTTP, WebSocket, TCP and Twitter, then capture and store that data. You can also use it to implement simple Representational state transfer (REST) APIs. You can do all of this without having to program much at all.

With the expansion and adoption of IoT solutions around the world, Node-RED has become an invaluable tool for solution architects and developers because of its ease-of-use in flow-based programming and solution mapping. The beauty of Node-RED is that almost anyone can quickly learn to use it – it is not limited to the realm of programmers. It can be used just as easily on a Raspberry Pi as it can in cloud environments such as IBM Bluemix.

13 INTERNET OF THINGS SECURITY

The IoT affects all aspects of our lives. We use the IoT in our homes, offices, and cars. This use has enabled us to do things we have not been aware of before, but there is a disadvantage with the IoT. It has become an increasingly attractive target for cyber criminals. Multiple connected devices mean more attack angles, which provide more opportunities for attackers. This has caused a security issue, and unless we do something to tackle this ever-growing problem, it will have serious consequences.

Security is a problem with the IoT. Sensors collect in many cases sensitive data, such as what people say and do in their own home. This means that safety is important when using the IoT, but until now IoT safety has been very poor. Many IoT embedded systems have been designed with little thought for basic security.

The IoT bridges the gap between the digital and the physical world, which means that hacking devices could have serious consequences in the real world. If hackers manage to take control of a car without a driver, it could end in an accident. Hacking sensors that control the temperature of a nuclear power plant could lead the operators to make a wrong and catastrophic decision.

Software errors have made smart home appliances such as refrigerators, ovens, dishwashers, and webcams open to hackers, who can use these items as an entry point to a network or to send spam. As the cost of creating smart IoT things gets smaller, these issues will only become more and more widespread and difficult to handle.

Errors are constantly being discovered in software, but many IoT devices lack the ability to be upgraded, which means they will be in constant danger. Attacks from hackers are now targeting IoT devices such as routers and webcams because these devices are deficient in security, making them easy to attack.

It has been demonstrated that cars connected to the Internet can be attacked and hackers can perform any number of unfortunate activities, including taking control of the entertainment system as well as unlocking or closing the doors while a car is moving. Another cause for concern is the hacking of medical devices. This phenomenon could have harmful and possibly fatal consequences for patient health.

There are increasing numbers of connected IoT devices; hence, there are more attack angles and more opportunities for hackers to attack. Security experts have warned of the potential risk of the large number of unsecured devices that have been connected to the Internet.

The idea of linking IoT devices and other objects to the Internet is relatively new. Therefore, safety has not always been considered when designing IoT products, which are often sold with old and unpacked embedded operating systems and software. Furthermore, buyers often fail to change the default passwords of smart devices, or if they change them, they do not use sufficiently strong passwords. To improve security, an IoT device that needs to be directly accessible over the Internet must be segmented into its own network and given limited access to the network. The network segment must then be monitored to identify potential irregular traffic, and something must be done immediately if there is a problem. IoT weaknesses open new opportunities for hackers. The vulnerabilities found on IoT devices are serious and have made IoT security a problem that needs to be resolved quickly.

13.1 SOME THREATS TO THE INTERNET OF THINGS SECURITY

The news about novel technology usually includes stories about hacking businesses, stolen identities, and hijacking app-connected cars. Security risks have significantly increased with the rise of connected IoT devices. Three types of IoT attacks are take control, steal information, or disrupt service (Figure 13.1).

Security Risks	
Take Control	Controls in homes for smart door locks and lighting systems can be vulnerable. Door locks in cars can be opened remotely.
Steal Information	Personal fitness devices can tell a hacker where you are. IoT devices in your home can give information about your personal life.
Disrupt Service	Hacked vehicle control systems can allow remote control of brakes. Peacemakers can be attacked remotely.

Figure 13.1 Three types of attacks on the Internet of Things (IoT)

IoT attacks are increasing and there are several reasons why hackers attack IoT devices (Figure 13.2).

Reasons for Attacks	
Lack of Security Software on the Devices	In contrast to regular computers, IoT devices do not have a firewall or virus scanner.
Multiple Devices with the Same Security Mechanisms	Once an attack works with one device it will work with thousands.
Less Experienced Device Producers	The businesses usually come from an industry vertical and often lack the information technology security expertise of server/computer manufacturers.
Devices Are Out of Reach	Device owners deploy their equipment remotely. Often an owner will not realise that the devices have been compromised until it is too late. Once an attacker has control over a device, it could run all day long before being physically shut down by the owner.

Figure 13.2 Why hackers are targeting Internet of Things (IoT) devices

A hacker is an individual who uses computer, networking, or other skills to overcome a technical problem. The term may also refer to anyone who uses their abilities to gain unauthorised access to systems or networks to commit crimes (Figure 13.3).

Attackers	
Amateur Hackers and Script Kiddies	Their objective is usually fame among their peers, either by targeting a high-profile victim or by demonstrating an ability to infect many devices in a single attack.
Criminal Businesses	These organizations take advantage of vulnerabilities within the target to generate revenue for themselves.
Governments/Intelligence Organisations	Acting in the safety of their citizens, intelligence agencies attempt to secure access to important information.
Political Interest Groups	They attack organizations that they think are morally corrupt.

Figure 13.3 Some types of Internet of Things (IoT) attackers

Each type of attacker can have different talents and aims, either individually or group based. Given the same tool, different classes of attackers could achieve different outcomes.

Cyber espionage is a form of cyber-attack that steals classified, sensitive data or intellectual property to gain an advantage over a competitor or government entity. Cyber espionage groups with large resources and highly qualified petty criminals are the most common type of IoT attacker. In many cases, they have developed advanced malware with the ability to mutate and to avoid detection in IoT networks, or they exploit distributed denial-of-service (DDoS) attacks as a means of extortion.

13.2 SOME INTERNET OF THINGS SECURITY ATTACKS

Physical Attacks

Physical attacks occur when IoT devices can be physically accessed by anyone. With most cybersecurity attacks occurring from the inside of a company, it is essential that your IoT devices are in a protected area, which is often not an option. Many physical cybersecurity attacks begin with the assailant inserting a USB drive to spread malicious code. Hence, it is more important than ever to add AI-based security measures to ensure your devices and data are protected.

Encryption Attacks

When an IoT device is unencrypted, the intruder can sniff the data and capture it for use later. In addition, once encryption keys are unlocked, cyber-assailants can install their own algorithms and take control of your system. For these reasons, encryption is a must-have in the IoT environment as part of your cyber security efforts.

Firmware Hijacking

If you are not keeping up with your IoT firmware updates, you are at risk of a cyber security attack. Be sure to check that your updates are from the expected source; otherwise, an attacker could hijack the device and download malicious software. Something else to keep in mind is that most hardware makers do not cryptographically sign embedded firmware.

Botnets

A bot is a software application that is programmed to do certain tasks. A botnet comprises several Internet-connected devices, each of which runs one or more bots. Botnets can be used to perform DDoS attacks, to steal data, to send spam and to allow the attacker to access the device and its connection.

Mirai is a malware that turns networked devices running Linux into remotely controlled bots that can be used as part of a botnet in large-scale network attacks. It primarily targets online consumer devices such as IP cameras and home routers. Mirai continues to be a problem today with millions of IoT devices affected.

Denial of Service Attack

A denial-of-service (DoS) attack occurs when a service, such as a website, becomes unavailable. Many systems attack one target through a botnet, which forces many devices to request a service at the same time. While attackers, in this case, are not typically aiming to capture data, they are seriously impacting a business if services become unavailable.

Man-in-the-Middle Attack

A man-in-the-middle attack occurs when a hacker breaches communication between two separate systems. By secretly intercepting communications between two parties, this type of attack tricks the recipient into thinking they are receiving a legitimate message. In other words, the man in the middle begins communicating with both parties, hence the name. It might look like an email from your bank, requesting that you log in to perform a task. Now, the attackers' fake website gathers your credentials, so the attacker can inflict further damage.

Ransomware

Ransomware is a type of malware that locks down access to files by encrypting them. Then, the attackers sell you the decryption key so that your files can be accessed again. Naturally, this type of attack can disrupt day-to-day business and the encryption key often comes at a hefty price. Imagine if hackers were able to access a power grid and refused to give the keys back for days. Cue the blackout.

Eavesdropping

In this type of attack, a hacker intercepts network traffic to steal sensitive information via a weakened connection between an IoT device and a server. Eavesdropping is typically done by listening to digital or analogue voice communication or via the interception of sniffed data. Again, in this case, the attacker walks away with sensitive, corporate data.

Privilege Escalation

Hackers look for IoT device bugs and weaknesses to gain access to resources that are typically protected by an application or user profile. In this type of attack, the hacker seeks to use their newly gained privileges to deploy malware or steal confidential data.

Brute Force Password Attack

In this scenario, hackers submit many passwords or passphrases with the hope of guessing the correct one, providing them access to your IoT devices. Alternatively, they use software to generate many consecutive guesses. Now that the attacker has access to your device, they could install malware or steal business-critical data.

13.3 SECURITY CHALLENGES

Any distributed system is expected to be well secured and reliable and meet the clients' privacy criteria. Moreover, when the issues involve the defence or medical sector, the security system is the first thing to be considered. IoT systems have vast applications in sensitive government and personal lives. However, we never can say that the environment is fully secure.

The key challenges in IoT security are:

1. Most of the IoT devices are extremely small. It is very hard to add an extra security module to those tiny things.
2. Most of the things have very low computational capabilities, so many complex security algorithms are not suitable for them.
3. Limited power in IoT things is the most challenging barrier to IoT security. The extra security module – whether software or hardware – requires extra energy. However, IoT systems, especially wireless systems, are always expected to be energy efficient.
4. The software of things, in most cases, cannot be updated. So, the present secure device might become insecure after a few years.
5. The industry has not taken the physical layer security issues seriously until now. However, the hardware level threats are increasing exponentially in electronic devices.

Every IoT device provides privacy and security issues. These issues range from hackers who steal our data and even threaten our lives, to how businesses can easily acquire private data.

Because the progress of the IoT will not stop, these are major issues that consumers and businesses must consider before using devices connected to the Internet.

Security and privatisation are critical issues facing the development of the IoT. We will look at some of the challenges that are central to making the IoT safer.

More Devices Mean More Hacking Opportunities

The basic security issue of the IoT is that the number of devices increases behind the network firewall. At the beginning of the computer age, we only had to worry about protecting our computers. Later, we also had to worry about protecting our mobile phones. Now we must worry about protecting the car, our household appliances, our wearables, and a host of other IoT devices.

Because there are so many devices that can be hacked, hackers can achieve more. You may have heard about how hackers potentially control cars remotely. However, hackers can even use seemingly insignificant IoT devices such as baby monitors or your thermostat to reveal private information or just ruin your day. The point is, we need to think about what a hacker can do with any entity if they can break through security.

Need for Updates

Although you might be good at properly configuring a connected IoT device, there are other holes. Manufacturers of connected devices are often slow in generating updates – some companies do not provide support at all. Instead of updating previous devices, they prefer to fix security issues with the next version they make of the thing. Security and privacy on the IoT are therefore the responsibility of the user.

As the IoT is growing, we must worry about protecting more and more devices. However, even if you start to take security seriously, it is the technical companies that make these new units that are responsible. In addition, the problem is that these companies do not update their devices well enough or at all. This means that an IoT device that was safe when you bought it could become insecure after hackers discover new security holes.

Computers used to have this problem, but automatic and easier updates have helped alleviate the issue. Computers have automatic updates, in part because most users are too lazy to perform even the basic steps needed to keep the computer safe. Moreover, when you consider that the protection of the myriad IoT devices will be much more difficult than a single computer, this problem will be even worse.

Manufacturers Can Be a Security Threat

Hackers are scary, but they are far from the only threat to the IoT. Backdoors are discovered in software. In fact, companies that make and distribute interconnected devices can also use these devices to acquire personal data, which is particularly dangerous for money transfers.

Manufacturers face enormous pressure to get their IoT units out on the market, and they often end up compromising on security. Although they can offer firmware upgrades for some time, they often stop this endeavour and begin focusing on building the next device, giving customers a bit of outdated hardware that could become a security risk.

Privacy and the Internet of Things

The fact that sensors collect data on everything you do means that the IoT is a potential headache in terms of privacy. For example, smart homes can tell when you wake up, when your smart coffee machine is activated, how well you brush your teeth thanks to your smart toothbrush, which radio station you are listening to thanks to your smart speaker, the type of food you eat thanks to your smart oven or fridge, what the kids are thinking thanks to their smart toys and who visits you and walks past your house thanks to door cameras.

The safety of the IoT data is important for privacy. It is surprisingly easy to find out much about a person from a few different sensor readings. In one project, a researcher found that by analysing data mapping of just home energy consumption, carbon monoxide and carbon dioxide, temperature, and humidity all day they could figure out what someone had for dinner.

Consumers need to understand the safety risks of using IoT devices, and whether they are happy with it. Some of the same issues apply to businesses. For example, would your management team like to discuss a merger in a meeting room equipped with smart speakers and cameras? A recent study found that four out of five companies could not identify all the IoT units on the network.

Physical and Hardware Security

The security issues in the IoT are not bounded within data authentication, access control, client privacy and other attacks like data leakage. Hardware-level insecurity is also grabbing the attention of researchers and becoming a growing problem day by day. To get a complete hardware-secured IoT system, we need to secure the integrated circuits in the IoT-enabled devices. Many IoT deployments will be in remote and isolated areas, leaving sensors and edge routers vulnerable to physical attack. It does not help if data security is very good if someone can get access to or steal sensors and equipment.

13.4 HOW TO SECURE INTERNET OF THINGS DEVICES

The IoT has become an essential part of daily life and modern business operations. However, as the network of smart devices grows in number and complexity, it has become increasingly apparent that the risks they bring about can no longer be ignored in favour of all their benefits.

During the past years, the IoT has attracted attention due to its weak security. Close to 70% of organisations have suffered from an IoT-sourced cyberattack worldwide. The numbers are only expected to grow as IoT use increases and hackers become more familiar and comfortable with the technology.

There are countless ways you can secure your IoT network devices and minimise the risks. Some are more complex than others because they provide different levels of security. Whether you implement all suggested security measures should depend on the threat and whether the risk is greater than the effort. The measures described below are ordered from the least complex to more intensive solutions.

Keep Software Up to Date

Software updates are not just for a slimmer user interface or additional features. They often fix old bugs and patch security vulnerabilities. Failing to keep IoT devices' software up to date always leaves your network vulnerable to attacks, especially if they are connected to the Internet often.

Skip Defaults and Double-Down on Passwords

The least you can do to ensure your IoT network and devices remain secure is setting strong passwords and usernames different from the default mode. Regarding passwords, change them regularly every 30 to 90 days. If you have a complex network with numerous devices, utilise a password manager to avoid using similar or straightforward passwords, which puts you at a greater risk.

When you first add a device to your existing network, go through its settings and customise it to your exact needs, disabling features you do not use. Most modern IoT devices connect with networks and other devices when they are in close vicinity. While that might be beneficial within an active office environment, it also creates a security gap, where an unauthorised user could easily connect to your network.

Encrypt Your Online Connection

There are multiple ways you can secure your Internet connection by encrypting the data exiting your internal network and disguising any sensitive information or vulnerabilities. One of the simplest ways to do it is using a virtual private network (VPN). A VPN changes a device's IP address and encrypts all the data leaving it, creating an additional layer of security. Most VPNs also offer a kill-switch feature, where it kills your Internet connection if the VPN crashes. That feature ensures your IoT device never contacts the open Internet unprotected.

Secure Internet Connection

The router is the gateway between your IoT devices and network to the open Internet. Left unsecured, it is an easy opening that outsiders could exploit. For routers, replace the default network name, admin username and password with secure alternatives and change them regularly.

Make sure to set the highest level of encryption your router has. If it only supports weaker levels, consider upgrading to a newer router that supports Wi-Fi Protected Access 2 (WPA2) encryption. You can also take Internet security one step further by creating separate networks for your IoT devices and personal devices.

Use a Monitoring System

In environments that host elaborate networks of IoT devices and use them to run critical operations and analyse sensitive data, it is essential to keep a close eye on the state of the devices and data flow. A monitoring system tracks your devices' health and sends out alerts if anything is out of the ordinary: unusual data flow, suspected unauthorised access or connection to the Internet and other devices in the network, among other situations.

Utilise Network Segmentation

Network segmentation is the process of splitting an internal network into multiple, separate sub-networks. While the segments can communicate on occasion, they are generally independent and isolated from each other. A flat network, where all devices are connected – including IoT and employee devices – are often protected by a firewall or endpoint protection and detection software.

In the case of a successful cyberattack, the attacker gains access to the entirety of the network. Network segmentation prevents that by minimising damage and limiting the attack area. It also allows you to focus on limited security resources on segments with the most critical data.

Focus on Flexibility and Scalability

Regardless of the different security measures you decide to implement, it is important to consider scalability and flexibility, especially with the more complex security solutions. Switching to a smaller or larger network of the IoT is inevitable as needs and budgets vary. Having a flexible security system in place will allow for a smoother and safer transition when needed.

Use Multi-Factor Authentication

Multi-factor authentication (MFA) is your backup if your password fails, either because it was in a data leak or an attacker figured it out through brute force. Like other security measures, the complexity and number of authentication steps you add should correlate to your threat model and the level of security you are after.

MFA can be as simple as receiving a text message with a unique code every time you log in or generating a code natively on your smartphone, which are options for two-factor authentication. Higher MFA levels include physical authentication, where you need to insert a physical key, usually a USB stick, to log in. In high-security networks, MFA includes biometrics or verifying the time and geographical location before allowing you to log in.

14 DESIGN FOR THE INTERNET OF THINGS

The IoT has a large market: everything from cars to refrigerators can be connected to the Internet. However, few companies have the expertise or resources to build all parts of an IoT design from scratch. Online providers offer a variety of solutions to help designers add wireless connectivity to new or existing products. Developers, nevertheless, often hesitate to use new technologies, as they fear that the company lacks experience in the area and are uncertain about which technologies they should focus on.

A procedure called virtual prototyping is good way to start an IoT project. Virtual prototyping allows designers to make the right technology choices, and to define and test system attributes. A prototype is a preliminary version of a product. The purpose of a prototype is to demonstrate and to test function and design. In this way, cost limitations, time to marketing and performance requirements can be evaluated and balanced to ensure that the product is both valuable and affordable before making major investments in design or production.

14.1 UNDERSTAND THE APPLICATION

For IoT design to be successful, one must consider a multitude of different elements. A designer must understand fully all technological elements of the IoT, how to build a successful prototype, how to design properly with AI, how to anticipate design flaws and failures and how to design for the people who will use each system and/or product. We will touch upon each of the above points to help provide a better understanding of the complexities of designing for the IoT.

While engineers can connect different components to an IoT product, expertise is required to take all the necessary steps to get an IoT product on the market. It is important to understand fully the hardware, software and certification requirements of the project before beginning the design process. There are several important questions to ask in this analysis phase.

What Is the Aim of the Project?

For what should the project be used? Does it need to work in real time? Should it be used for automation or control? A good understanding of how a project works and for what the project will be used allows you to determine the required power and performance levels.

What Are the Requirements for Size?

It is often desirable that IoT units be as small as possible, but this endeavour can quickly become expensive. Take for example wearables. The performance of these portable IoT devices is often limited and they are only capable of supporting small amounts of data. If small size and high performance is the goal, a long-life battery will be needed to meet power consumption, which can quickly make the overall solution bigger and more expensive.

What Are the Requirements for the Communication Distance?

Inside a house or in a city with easily accessible Wi-Fi, range can be measured in feet or metres. However, outdoors or in a rural area, the range required may be several kilometres for a signal to reach the nearest server or wireless gateway, requiring a mobile or GPS interface. If there is a long transmission distance, higher power and higher frequencies will be needed. If the location is remote and cannot be reached easily, the battery life will be important. Disturbances with physical obstructions or other RF devices can also affect the operating distance.

What Is the Power Source?

The power source is a critical component in designing an IoT application because it affects both the communication distance and the battery life. Indeed, the longer the range, the more power is needed, and the more power is required, the shorter the battery life. If the device is to be powered by batteries alone, it is only necessary to design with the intention of saving power. Typically, a lithium-ion battery is the standard form of power in mobile devices due to its energy density.

Batteries lose capacity with many charge–discharge cycles. This capacity loss is expressed as a measure of the initial capacity, for example 30% loss after 1000 cycles. The rate of loss depends on chemistry and temperature. Typically, a lithium-ion battery can last for 10 years, while an alkaline battery will only last for 5 years.

Many network technologies will not work well with battery power. The frequency of the communication will also influence the choice of power supply.

Environmental Considerations

One of the many benefits of wireless systems is that they can often work out where people cannot, including inaccessible or dangerous environments. However, it is important to control the types of wireless systems that can operate in specific environments. You must take into consideration whether the environment is hot, cold, wet or dry. For example, an IoT device to be used in a freezer or a device used in a high-heat location will likely require more frequent monitoring and possibly a built-in emergency alert.

Communication

Should the device we make be able to communicate with other devices? In that case, one must ensure that they can cooperate. This goes beyond adhering to standards from organisations such as IEEE, ISO, and others, because even these known standards are sometimes open to interpretation in more than one way.

Safety

Users of IoT products want to know that their very personal information is safe and secure. It is important for designers to keep security issues and user privacy in mind. Users want to trust the product they are using and know that their information will be stored safely and securely.

14.2 PRINCIPLES OF GOOD INTERNET OF THINGS DESIGN

Designing an IoT solution comes with a set of entirely new design challenges. Consider that IoT systems usually consist of multiple elements such as physical device sensors, actuators, interactive devices, the network that connects these devices as well as the data gathered and then analysed to create a meaningful experience. We are also dealing with the physical context in which the user interacts with the solution. For example, for smart home devices, it can be your kitchen. That is why IoT implementations require various types of design, from industrial product design to service and business design. All these factors impact the total user experience on the IoT system. Hence, designing in this context is sometimes quite challenging.

Here are some tips for a good IoT design.

Safety First

Privacy and data security are the keys to good IoT design. Users should feel that their smart devices are secure and cannot be hacked, so they and their loved ones are not risking their privacy or a data breach.

Do Your Research

Designers of IoT devices and systems should look carefully into the users' problems and needs to deliver a better solution for their real-life problems. When getting into IoT design, you are no longer building products: you are building services and experiences that improve people's lives. In-depth qualitative research is the key to figuring out how you can achieve that goal. Assume the perspective of your customers to understand what they need and how your IoT implementation can solve their pain points. Research your target audience deeply to see what their existing experiences are and what they wish was different about them.

Make Good Use of Prototypes

IoT solutions are often difficult to upgrade. Once the user places the connected object somewhere, it might be hard to replace it with a new version, especially if the user would have to pay for the upgrade. Even the software within the object might be hard to update because of security and privacy reasons. Make sure that your design practices help to avoid costly hardware iterations. Get your solution right from the start. From the design perspective, it means that prototyping and rapid iteration will become critical in the early stages of the project.

Testing and Futureproofing

It is recommended to align the service lives of the IoT hardware and software elements. Once a connected object is integrated into a system, it becomes difficult to replace it with a newer version. This is also true for the object's software because it might be costly or challenging to update it due to security and privacy reasons. It is crucial to test the solution thoroughly to make it right from the beginning. Prototyping and rapid iterations of the hardware and complete system in the early stages of the project are essential. To prevent problems that might occur at the time of upgrades, it is also important to predict possible major changes.

15 INTERNET OF THINGS STATISTICS

15.1 THE SIZE OF THE INTERNET OF THINGS

The number of things connected to the Internet continues to increase steadily (Figure 15.1). The number of interconnected devices is set to explode in the subsequent years as Internet consumption rises and new gadgets and machinery hit the market. Everyday objects are increasingly connected to the Internet, ranging from smart refrigerators to smart toothbrushes. In the coming years, the number of smart units in our homes will only increase.

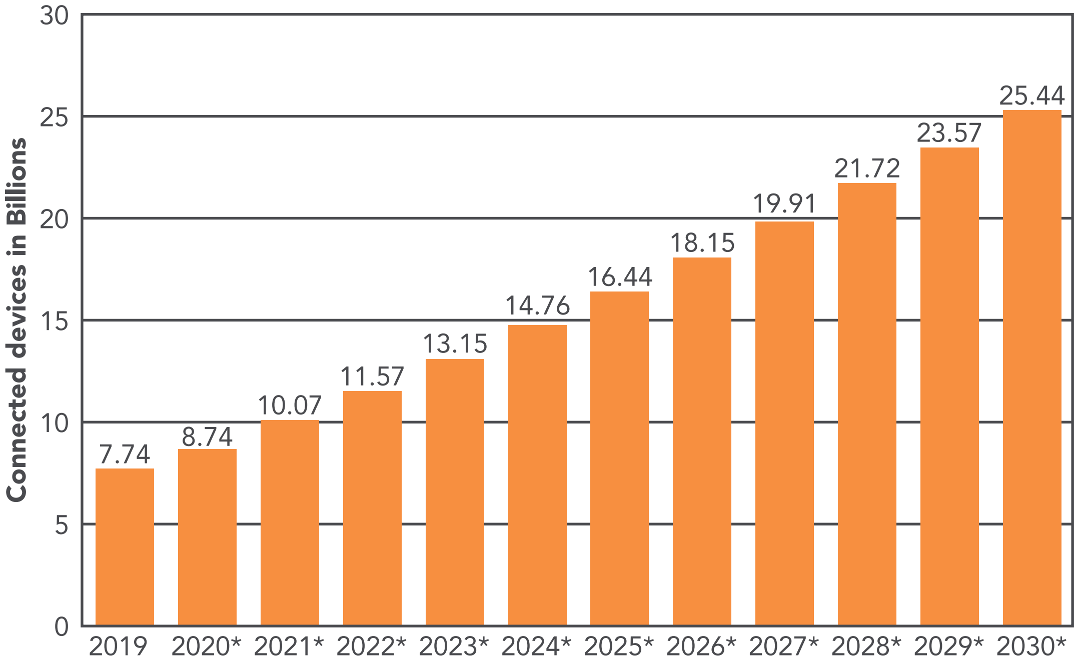


Figure 15.1 The number of Internet of Things connected devices worldwide from 2019 to 2030 in billions (Statista 2021)

IoT devices are becoming increasingly cheaper as the prices of sensors and communication via the Internet continues to fall. The sensors are getting smaller, better, and cheaper every year. The cloud that provides the necessary infrastructure is easily accessible and relatively affordable. This provides a good basis for IoT to grow in the years to come.

The possible uses of the IoT are endless. Anything or any device can become smart if it connects sensors to the device and it is connected to the Internet. Using software designed to analyse data and machine learning, the data from the sensors can provide valuable insights for people.

With so many opportunities, cheap infrastructure, and high demand, we can expect an explosion of connected IoT units in the coming years. It looks like we are moving towards a world that is fully automated.

15.2 WHAT IS THE FUTURE OF THE INTERNET OF THINGS?

The future of the IoT has the potential to be limitless. Advances to the industrial Internet will be accelerated through increased network agility; integrated AI; and the capacity to deploy, to automate, to orchestrate and to secure diverse use cases at hyperscale. This section is built on Gartner's predictions on the future for IoT. Gartner is the world's leading research and advisory company.

More Sensors

Embedded sensors are used to collect data from around the world. The IoT sensor market alone is expected to be worth a staggering USD 27 billion by 2022. Hence, we will see a myriad of sensors in the future.

The sensor market will evolve continuously. New sensors will enable a wider range of situations and events to be detected, current sensors will fall in price to become more affordable or will be packaged in new ways to support new applications and new algorithms will emerge to deduce more information from current sensor technologies.

Artificial Intelligence and Machine Learning

AI means that machines can perform tasks in a way that is intelligent. Machine learning is technically a branch of AI; it is based on the idea that we can build machines to process data and to learn on their own, without human supervision. Machine learning is going to become an integral part of IoT devices. Big companies like Microsoft, IBM and Google are investing in AI and machine learning. It is the only way to move forward.

The technological landscape for AI is complex and will remain so, with many IT vendors investing heavily in AI, variants of AI coexisting and new AI-based tolls and services emerging. Despite this complexity, it will be possible to achieve good results with AI in a wide range of IoT situations.

AI will be applied to a wide range of IoT information, including videos, still images, speech, network traffic activity and sensor data. High-profile examples of AI include autonomous vehicles (such as drones and self-driving cars), medical diagnosis, creating art (such as poetry), proving mathematical theorems, playing games (such as Chess or Go), search engines (such as Google search), online assistants (such as Siri), image recognition in photographs, spam filtering, predicting flight delays, predicting judicial decisions, and targeting online advertisements.

The Shift from Intelligent Edge to Intelligent Mesh

The shift from centralised and cloud to edge architectures is well under way in the IoT space. However, this is not the end because the neat set of layers associated with edge architecture will evolve to a more unstructured architecture comprising a wide range of things and services connected in a dynamic mesh. These mesh architectures will enable more flexible, intelligent, and responsive IoT systems, although often at the cost of additional complexities.

Trusted Hardware and Operating System

Gartner surveys invariably show that security is the most significant area of technical concern for organisations deploying IoT systems. This outcome is because organisations often do not have control over the source and nature of the software and hardware being utilised in IoT initiatives. However, hardware and software combinations are expected to be deployed that together create more trustworthy and secure IoT systems.

Silicon Chip Innovation

Currently, most IoT endpoint devices use conventional processor chips, with low-power ARM architectures being particularly popular. However, traditional instruction sets, and memory architectures are not well suited to all the tasks that endpoints need to perform. For example, the performance of deep neural networks (DNN) is often limited by memory bandwidth rather than processing power. It is expected that new special-purpose chips will reduce the power consumption required to run a DNN, enabling new edge architectures and embedded DNN functions in low-power IoT endpoints. This will support new capabilities such as data analytics integrated with sensors and speech recognition included in low-cost, battery-powered devices. This trend of enabling silicon chip functions such as embedded AI will enable organisations to create highly innovative products and services.

New Wireless Networking Technologies for the Internet of Things

IoT networking involves balancing a set of competing requirements, such as endpoint cost, power consumption, bandwidth, latency, connection density, operating cost, quality of service and range. No single networking technology optimises all these factors, and new IoT networking technologies will provide additional choice and flexibility. They should explore 5G, the forthcoming generation of low Earth orbit satellites and backscatter networks.

Tightened Security and Higher Physical Standards

Hacks and data breaches have proved that the security of IoT devices needs to be much better. Another change we will see is in the physical standards. Devices will be valued based on their level of security and endurance in the future rather than on how cheap they are to produce.

Data Privacy Will Become a Priority

When it comes to IoT, one of the main concerns consumers and developers have had concerns privacy. IoT products are going to be strictly regulated, and privacy will be the primary concern in the future.

5G Networks Will Continue to Fuel Growth of the Internet of Things

Major wireless carriers will continue to roll out 5G networks, which promise greater speed and the ability to connect more smart devices at the same time. Faster networks mean the data accumulated by your smart devices will be gathered, analysed, and managed to a higher degree. That will fuel innovation at companies that make IoT devices and boost consumer demand for new products.

More Cities Will Become Smart

Consumers will not be the only ones using IoT devices. Cities and companies will increasingly adopt smart technologies to save time and money. Hence, cities will be able to automate, to manage remotely and to collect data through things like visitor kiosks, video camera surveillance systems, bike rental stations and taxis.

Cars Will Get Even Smarter

The arrival of 5G will shift the auto industry into a higher gear. The development of driverless cars, as well as the connected vehicles already on the road, will benefit from data moving faster. You might not think of your car as an IoT device, but new cars will increasingly analyse your data and connect with other IoT devices, including other high-tech vehicles on four wheels.

BIBLIOGRAPHY

Agarwal, Tarun. Embedded Systems Role in Automobiles with Applications. Retrieved January 2019, from https://www.edgefx.in/importance-of-embedded-systems-in-automobiles-with-applications/

Bennett, Eleanor. (2021). How to Secure IoT Devices. Retrieved July 2021, from https://www.iotforall.com/how-to-secure-iot-devices-2

Bose, Anjana. Embedded System – Characteristics, Types, Advantages & Disadvantages. Retrieved January 2019, from https://electricalfundablog.com/embedded-system-characteristics-types-advantages-disadvantages/

EMnify. (2020), IoT Attacks, Hacker Motivations, and Recommended Countermeasures. Retrieved July 2021, from https://www.iotforall.com/iot-attacks-hacker-motivation

Fuller, JR. (2016). The 4 Stages of an IoT Architecture. Retrieved March 2019, from https://techbeacon.com/enterprise-it/4-stages-iot-architecture

Gartner. (2018). Gartner Identifies Top 10 Strategic IoT Technologies and Trends. Retrieved May 2019, from https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends

Gregersen, Carsten. (2021). A Complete Guide to IoT Protocols & Standards In 2021 Retrieved June 2021, from https://www.nabto.com/guide-iot-protocols-standards/

Gyarmathy, Kaylie. 5 IoT Statistics You Need to Know in 2019. Retrieved May 2019, from https://www.vxchnge.com/blog/iot-statistics

Hasan, Mehedi. (2012). Top 15 Best Embedded Systems Programming Languages. Retrieved January 2019, from https://www.ubuntupit.com/top-15-best-embedded-systems-programming-languages/

Hegdes, Melissa. (2018). Cloud Storage VS Local Storage – Which is Right for Your Business? Retrieved May 2019, from https://www.contegix.com/blog/cloud-storage-vs-local-storage-which-right-your-business

i-scoop.eu. IoT Platforms - IoT Platform Definitions, Capabilities, Types, and Market. Retrieved July 2021, from https://www.i-scoop.eu/internet-of-things-guide/iot-platform-market-2017-2025/

Jindal, Taru. Role of Embedded System in Robotics. Retrieved January 2019, from http://www.robogalaxy.com/post/Role-of-Embedded-System-in-Robotics

Jovanović, Bojan . (2021). 45 Fascinating IoT Statistics for 2021 | The State of the Industry. Retrieved July 2021, from https://dataprot.net/statistics/iot-statistics/

Kaa. What Is an IoT platform? Retrieved April 2019, from https://www.kaaproject.org/what-is-iot-platform

Karnaukh, Rina. Designing the IoT: Best Practices and Examples. Retrieved July 2021, from https://onix-systems.com/blog/designing-the-internet-of-things-best-practices-and-examples

Koley, Subha and Ghosal, Prasun. (2015). Addressing Hardware Security Challenges in Internet of Things: Recent Trends and Possible Solutions. Retrieved May 2019, from https://www.researchgate.net/publication/281150039_Addressing_Hardware_Security_Challenges_in_Internet_of_Things_Recent_Trends_and_Possible_Solutions

Lea, Perry. (2018). Internet of Things for Architects. Pact Publishing.

Lee, Jeffrey. (2018). How Mesh Networking Will Make IoT Real. Retrieved May 2019, from https://hackernoon.com/how-mesh-networking-will-make-iot-real-b5b88baab63b

McClelland, Calum. (2020). What Is an IoT Platform? IoT for All. Retrieved January 2019, from https://www.iotforall.com/what-is-an-iot-platform/

Noergaard, Tammy. (2013). Embedded Systems Architecture. A comprehensive Guide for Engineers and Programmers. Newnes

Norton. (2019). The Future of IoT: 10 Predictions about the Internet of Things. Retrieved July 2021, from https://us.norton.com/internetsecurity-iot-5-predictions-for-the-future-of-iot.html

O'Connor, Chris. (2016). Wiring the Internet of Things. Retrieved January 2019, from https://www.ibm.com/blogs/internet-of-things/wiring-internet-things/

ONE Tech. (2019). 10 Types of Cyber Security Attacks in IoT. Retrieved July 2021, from https://www.onetech.ai/en/blog/10-types-of-cyber-security-attacks-in-the-iot

Prokopets, Maryna. (2018). Ultimate List of 30 IoT Platforms You Must Try in 2018. Retrieved June 2019, from https://dzone.com/articles/ultimate-list-of-30-iot-platforms-for-your-iot-pro

Reynolds, Ian J.H. (2020). IOT Architecture: 3 Layers, 4 Stages Explained. Retrieved June 2021, from https://www.zibtek.com/blog/iot-architecture/

Sakowich, Natallia. 8 Most Popular IoT Protocols and Standards You Need to Know. Retrieved June 2021, from https://www.sam-solutions.com/blog/internet-of-things-iot-protocols-and-connectivity-options-an-overview/

ScienceDirect. (2012). Embedded Systems Design. Retrieved January 2019, from https://www.sciencedirect.com/topics/computer-science/embedded-system-design

Senseware.co. The Truth About IoT Implementations - Wireless vs. Wired. Retrieved June 2021, from https://blog.senseware.co/2017/10/10/iot-implementations-wireless-vs-wired

Sethi, Pallavi and Sarangi, Smuruti R. (2017). Internet of Things: Architectures, Protocols, and Applications. Retrieved April 2019, from https://www.researchgate.net/publication/312957467_Internet_of_Things_Architectures_Protocols_and_Applications

Singh, Hemendra. (2018). Statistics That Prove IoT will become massive from 2018. Retrieved March 2019, from http://customerthink.com/statistics-that-prove-iot-will-become-massive-from-2018/

Sonee, Sapna. (2021). Top IoT Communication Protocols Updated 2021 - ZigBee, NFC, And More. Retrieved June 2021, from https://hashstudioz.com/blog/top-iot-communication-protocols-2020/

Southwest Center for Microsystems Education (SCME), University of New Mexico. (2011). Introduction to Transducers, Sensors, and Actuators. Retrieved January 2019, from http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/More%20on%20Transducers%20Sensors%20and%20Actuators.pdf

Statista. (2019). Internet of Things - Statistics & Facts. (2019). Retrieved March 2019, from https://www.statista.com/topics/2637/internet-of-things/

Szczęsny, Jakub . (2020). IoT Design: These Are the Principles You Have to Remember. Retrieved July 2021, from https://concisesoftware.com/iot-design-principles/

Tutorialspoint. (2019). Internet of Things (IoT) Tutorial. Retrieved March 2019, from https://www.tutorialspoint.com/internet_of_things/

Tutorialspoint. (2019). Embedded Systems Tutorial. Retrieved January 2019, from https://www.tutorialspoint.com/embedded_systems/index.htm

Vredenberg, Loek. (2020). 5G og Edge Computing (in Norwegian). Retrieved June 2021, from https://www.ibm.com/blogs/think/no-no/2020/06/11/5g-og-edge-computing-hva-skal-vi-med-det/

Wopata, Matthew. (2021). 5 Things to Know About the IoT Platforms Market. Retrieved July 2021, from https://iot-analytics.com/5-things-to-know-about-iot-platforms-market/

Wikipedia. Embedded System. Retrieved 2019, from https://en.wikipedia.org/wiki/Embedded_system

Wikipedia. Internet of Things. Retrieved April 2019, from https://en.wikipedia.org/wiki/Internet_of_things